Solution of the Incompressible Navier-Stokes Equations on Unstructured Meshes

by

David J. Charlesworth

Submitted for the Degree of Doctor of Philosophy

The Department of Mechanical Engineering

University College London

August 2003

ABSTRACT

Since Patankar first developed the SIMPLE (Semi Implicit Method for Pressure Linked Equations) algorithm, the incompressible Navier-Stokes equations have been solved using a variety of pressure-based methods. Over the last twenty years these methods have been refined and developed however the majority of this work has been based around the use of structured grids to mesh the fluid domain of interest.

Unstructured grids offer considerable advantages over structured meshes when a fluid in a complex domain is being modelled. By using triangles in two dimensions and tetrahedrons in three dimensions it is possible to mesh many problems for which it would be impossible to use structured grids. In addition to this, unstructured grids allow meshes to be generated with relatively little effort, in comparison to structured grids, and therefore shorten the time taken to model a particular problem. Also, through the use adaptive refinement, the mesh generation process can be coupled to the solution algorithm to allow the mesh to be refined in areas where complex flow patterns exist.

Whilst the advantages to unstructured meshes are obvious they have inherent difficulties associated with them. The computational overheads of using an unstructured grid are increased and the discretisation process becomes more complex. Also, it is inevitable that some of the discretisation methods used as standard on structured grids, do not perform as accurately when used on an unstructured mesh. Therefore, the use of unstructured meshes in computational fluid dynamics (CFD) is still an area of active research.

This thesis aims to investigate the use of unstructured meshes to solve the incompressible Navier-Stokes equations using the SIMPLE algorithm. A discretisation strategy drawing on the work of others is developed, that attempts to maintain the accuracy of the solution despite the discretisation problems that unstructured grids present. Particular attention is paid to the convective term in the momentum equations, which is often the cause of inaccuracy in pressure-based

solvers. High order convective models, first developed for structured meshes, are adapted for use within an unstructured discretisation to ensure stable and bounded solutions are calculated. To reduce computational costs, the discretisation is based on a pointer system that aims to minimise the amount of connectivity data stored for a particular grid. In addition an efficient multigrid algorithm accelerates the solution of the equations to achieve more realistic calculation times.

As an initial test of the solver's accuracy and efficiency, calculated results are compared with standard laminar flow problems in both two and three dimensions. However, for any solution strategy to be of practical use it must be able to model turbulent flow. To that end the algorithm is extended to find solutions to the incompressible Reynolds averaged Navier-Stokes equations, using the k-ε turbulence model to close the equations. Again, two and three-dimensional problems are used to test the solver's accuracy and efficiency at calculating turbulent flow. Finally the findings of the research work are summarised and conclusions drawn.

CONTENTS

FIGURES

TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Mehrdad Zangeneh for his supervision and help with the research work that has lead to this thesis. Also, my thanks goes to the Department of Mechanical Engineering at University College London as a whole and in particular to Dr. Kevin Drake for his support and belief in the work presented here. For their financial support I would like to thank the Engineering and Physical Sciences Research Council, as without their assistance this thesis would not have been written. Finally, I would like to thank my family and friends for all their support in getting to me to this stage, it's been an uphill struggle but hopefully it is now coming to a conclusion.

CHAPTER 1

1 Introduction

Computational fluid dynamics (CFD) has been an analysis tool used by the engineering community for almost thirty years. In it's infancy CFD was initially applied to problems in aerodynamics, however it's merits were quickly realised and it was soon being employed throughout the engineering disciplines, to a wide variety of problems. The advantage CFD gives to an engineer is the ability to predict with reasonable accuracy the details of complex flow patterns that influence practical engineering problems. This is achieved without performing model testing in wind tunnels, towing tanks or similar experimental facilities. Instead CFD utilises the power of digital computers, to solve the governing equations of fluid dynamics through numerical rather than analytic techniques. Clearly, numerical calculations can be performed on a digital computer more quickly and with much less manpower than model testing. However, these calculations can only really be relied upon within the range of parameters and conditions within which they have been validated by experiment. Nevertheless, CFD offers a much more cost effective analysis tool than experimental testing. Also, the reduction in time to perform an analysis of a given fluid flow problem allows that problem to be studied more thoroughly, which in turn should result in improved engineering design. Whilst model testing is still and will remain an invaluable part of engineering, CFD has gradually been used in addition to and in some cases instead of conventional experimental analysis. Inevitably this trend will continue as the time and cost benefits of CFD are impossible to ignore, and if the advances in computing technology since the nineteen seventies continue these benefits will only be amplified.

CFD requires that the domain where the fluid flow of interest exists is split into a number of small cells through the use of a grid or mesh. This allows the governing equations of the given fluid flow to be discretised and thereby converted into a large system of simultaneous algebraic equations which can be solved numerically. The

first CFD techniques that were developed used structured orthogonal meshes to achieve this goal. This form of grid is typically constructed from squares or rectangles in two dimensions and cubes or bricks in three dimensions and obviously is relatively easy to construct. Also, it's line structure offers significant benefits when creating computer programs to solve the algebraic equations, that result from the discretisation process. However, whilst there are significant advantages for structured orthogonal meshes these are tempered by the limited number of domains that can be meshed using squares or cubes. In most engineering applications of practical interest fluid is found in complex geometries and therefore more sophisticated meshing techniques are required.

Initial attempts at finding grid generation techniques that could mesh more complex geometries still revolved around structured grids. Essentially the orthogonal constraint on the mesh was removed resulting in mesh generation techniques that were free to conform to more complicated shapes. Typical examples of this strategy are the H, O and C grids often found in turbomachinery and aeronautical applications. In some cases these new curvilinear meshes were used directly to discretise the governing equations, but more often they were transformed along with the governing equations to uniform orthogonal meshes before discretisation was performed. These new techniques allowed CFD to be used in a wider range of engineering applications, but they did not result in a general method that could be applied to any geometrical configuration.

In order to achieve this aim a complete rethink of how grids were generated was required. This resulted in the development of unstructured grid generation techniques that allow almost any geometrical configuration to be meshed. The majority of these techniques used triangles and tetrahedrons, in two and three dimensions respectively, as the basic element from which to construct meshes. Clearly this change of basic element forces the mesh to be unstructured, as maintaining any line structure is impossible with anything other than very regular triangles and not possible at all with tetrahedrons. However, it is also this change that allows the unstructured grid to mesh very complex geometries. In addition, the unstructured mesh also facilitates adaptive refinement as part of the solution process. This technique uses information from the solution found on a given mesh to create a new mesh that has areas of local

grid refinement where complex flow patterns exist. Clearly this method allows the governing equations to control how fine or coarse a mesh should be in a given region and therefore should lead to more accurate solutions. Both adaptive refinement and the unstructured grid's basic geometric flexibility, contribute to one other advantage over it's structured counterpart, that is a significant reduction in the time and effort required to generate meshes. Often in CFD calculations based on structured grids the mesh generation process contributes to more than fifty percent of the overall time taken to achieve a solution. Obviously any technique that reduces this time constraint is advantageous as ultimately it allows either more analysis of a given problem to be performed or reduces the cost of completing a specific study.

On the face of it the unstructured grid offers many of the features that would make up a good mesh generation strategy, but inevitably there are disadvantages to the technique. The foremost of these is the lack of line structure within the grid which introduces the computational overhead of having to store grid topology data. Also, structured discretisation techniques are not necessarily readily applicable to unstructured grids, and if they are, then they do not always perform with the required level of accuracy. Nevertheless these issues can be addressed, as will be shown in this thesis, and consequently as CFD continues to develop it is likely that unstructured meshes will become more prevalent.

The incompressible Navier Stokes equations govern the fluid flow in and around a multitude of engineering applications. Some typical examples from aeronautics and turbomachinery are low speed pumps, low Mach number airfoils and ship propellers. Consequently, a number of CFD techniques have been developed for the solution of these equations and these can be broadly split into schemes based upon finite volume or finite element discretisation methods. The finite element method was developed initially for structural calculations, although it was soon applied to fluid dynamics. Since then, there has been a large body of research dedicated to using this method with unstructured meshes to solve the incompressible Navier Stokes equations. See, for example, Thomasset[1], Gunzburger[2] and Lohner[3]. In contrast research into using the finite volume method to solve the incompressible Navier-Stokes equations has been largely based around structured grids. Nevertheless, the following authors have published relevant literature, Davidson[4], Mathur and Murthy[5], Thomadakis and

Leschziner[6], Despotis and Tsangaris[7], Jiang and Przekwas[8], de Foy and Dawes[9], Watterson[10], Weiss and Smith[11], and Anderson *et al.*[12] and a summary of their work is presented below.

One of the chief difficulties in solving the incompressible Navier-Stokes equations is finding a suitable equation to solve for the pressure field. The solution adopted by [4], [5], [6], [7], [8], [9] and [10] was to use a pressure correction method to resolve this issue. The methods they used took one of three forms, Hirt and Cook's[13] pressure correction scheme or Harlow and Welch's[14] marker and cell (MAC) method or in the majority of cases, Patankar's[15] Semi Implicit Method for Pressure Linked Equations (SIMPLE) or one of it's derivatives. The remaining authors used Chorin's[16] artificial compressibility condition, to recast the incompressible Navier-Stokes equations into a form that mimics their compressible counterparts, and therefore removes the difficulty of finding an equation for pressure. In addition, to the method by which pressure is found, the location at which variables are stored is also significant in unstructured incompressible solvers. [4], [5], [8], and [11] stored their variables at the centre of the grid cells whilst [9], [10] and [12] all opted for the cell vertices as the storage location. An alternative semi-staggered approach similar to Patankar's[15] was adopted by [6] and [7] to avoid odd-even decoupling of the velocity and pressure fields. This effect was avoided by the other authors through the use of a variety of schemes all of which removed the necessity for staggering and allowed them to collocate their variables in a given cell. [8], [9] and [11] all published methods that were designed to calculate unsteady solutions, whilst the remaining authors presented steady state methods. Finally, the published literature also differs on whether the methods presented are two or three-dimensional. In the case of [4], [6], [7], [8] and [11] the schemes were two-dimensional whereas [5], [9], [10] and [12] all presented three-dimensional solutions.

This thesis seeks to extend existing finite volume techniques that are steady, cell centred and based upon the SIMPLE algorithm. As indicated by the summary above methods of this nature have been published by Davidson[4] and Mathur and Murthy[5]. The difference between the two methods is the way in which the convective and diffusive terms in the Navier-Stokes equations are discretised. Davidson[4] adopts a central differencing approach to both terms whilst Mathur and Murthy[5] use a second

order upwinding scheme for the convective term and central differencing for the diffusive term. There is a subtle difference between Mathur and Murthy's[5] central differencing approach to the diffusive term and Davidson's[4]. This allows Mathur and Murthy[5] to calculate three-dimensional problems but in essence they are the same formulation. The convective term however is quite different. As Leonard[17] established central differencing is an unstable approach to discretisation of the convective term. To mitigate against this problem Davidson[4] added an artificial dissipation term to stabilise his central differencing formulation. In contrast Mathur and Murthy's[5] upwind formulation does not require this stabilisation because, as Leonard[17] showed, second order upwinding is intrinsically stable. However, both formulations are not easily extended to orders of accuracy higher than two. Leonard[17] showed that third order convective models on structured grids could prove advantageous and therefore the ability to use higher order convective models on unstructured meshes is desirable. Here, a formulation is developed that allows a second, third or higher order upwind discretisation of the convective term to be used. In addition, whilst Davidson's[4] and Mathur and Murthy's[5] methods could only calculate laminar flows, here, the k-ε turbulence model is introduced so that turbulent flows can also be calculated. The solution algorithm also utilises a form of algebraic multigrid specially adapted for unstructured meshes to solve the pressure correction equation. This allows solutions on both large and small meshes to be found within a reasonable time frame.

The research presented here is essentially divided into four sections. To begin with Chapter 2 presents a detailed literature survey. This covers different discretisation methods for unstructured grids, convective modelling, turbulence modelling and unstructured grid generation. Following this are two chapters that detail the methodology used here to solve the incompressible Navier-Stokes equations. Chapter 3 presents the discretisation procedure and solution algorithm whilst Chapter 4 gives details of how the algebraic multigrid method was adapted for unstructured grids. Chapters 5 and 6 are the results section of the thesis. Four laminar test cases are presented in Chapter 5 and this is followed by three turbulent test cases in Chapter 6. Finally, Chapter 7 draws conclusions about the work presented and suggests areas of further research that would develop the ideas presented.

CHAPTER 2

2 Literature Survey

2.1 Introduction

In order to give background to the development of a cell centred, finite volume, SIMPLE based algorithm, which facilitates the use of high order convective models for both laminar and turbulent flow, a survey of the open literature is presented here. To begin with an overview of discretisation methods suitable for unstructured meshes is presented. Following this will be background to the development of the SIMPLE algorithm and a review of the family of pressure correction algorithms that it spawned. This section will also discuss the problems of decoupling of velocity and pressure fields and the possible solutions to this problem.

The focus of the survey will then switch to the discretisation of the convective term in the momentum equations. As highlighted in Chapter 1, this term has caused considerable difficulty for the researchers in this field because standard methods of discretising the convective term have been either inaccurate or unstable. A number of ingenious solutions to this problem have been put forward although these have been based mainly on structured grids. Nevertheless, a discussion of these techniques provides an insight into the underlying problems associated with the discretisation of the convective term, and can therefore help when trying to solve the same problems on unstructured meshes.

Having discussed the pertinent issues associated with the discretisation of the governing differential equations and reviewed the solution algorithms used to solve the resulting algebraic equations, the attention of this literature survey turns to turbulence modelling. This area is still the subject of significant research and consequently a brief summary of the development of the field and the important issues facing this area of CFD is useful. Also, although not directly an area where research has been carried out for this thesis, the topic of grid generation cannot be

overlooked, as it underpins all solution algorithms particularly so for unstructured grids. Therefore an overview of this topic is presented before this chapter closes.

2.2 Discretisation Methods

Over the last twenty years the numerical solution of the incompressible Navier-Stokes equations has been dominated by two discretisation strategies, the finite volume[18] and finite element[19] methods. Neither method has been proven to be more accurate than the other, however as we have seen the application of the finite element method to unstructured grids has been much more prevalent than it's counterpart the finite volume method. The research documented in this thesis attempts to develop the finite volume method for unstructured grids but for completeness an overview of both techniques is presented in this section.

In order to compare the two methods consider the following Poisson equation

$$\nabla^2 u - f = 0 \text{ where } u = u(\mathbf{x}) \text{ and } \mathbf{x} \in V \qquad (2.1)$$

Adopting the finite element approach first, an approximate solution to equation (2.1) is sought, where the error between the exact and approximate solution is at a minimum. The most general way to accomplish this is to use the method of weighted residuals i.e.

$$\int_V W_i L(u^*) dV = 0 \text{ where } i = 1,2,\ldots\ldots\ldots,m \qquad (2.2)$$

where L is the operator on the left hand side of equation (2.1), u^* is the approximate solution and W_i is a set of weighting functions introduced to minimise the error. It is generally difficult to find an appropriate set of weighting functions to satisfy equation (2.2) across a complete domain, for all but the simplest of geometries. Nevertheless, this problem can be circumvented by splitting the domain of interest into a number of sub-domains or elements and applying equation (2.2) locally. These local equations can then be summed over all the sub-domains or elements to represent the complete integral i.e.

$$\sum_{elements} \int_{V_{elemens}} W_i L(u^*) dV = 0 \qquad (2.3)$$

The approximate solution, u^* is then assumed to take the following form

$$u^* = N_j(\mathbf{x}) u_j^* \qquad (2.4)$$

where j takes values from one to the number of discrete points making up an element i.e. three in the case of a triangle, u^*_j are the approximate values of u at those points and N_j are so called trial functions that can be chosen such that u^* assumes a particular distribution within an element, for example linear or quadratic. Substituting equation (2.4) into equation (2.3) yields

$$\sum_{elements} \int_{V_{element}} W_i \left[\nabla^2 \left(N_j u_j^*\right) - f\right] dV = 0 \qquad (2.5)$$

Furthermore, applying the divergence theorem and assuming that f can be expressed by an analogous expression to equation (2.4) gives

$$\sum_{elements} u_j^* \int_{V_{element}} \nabla W_i . \nabla N_j dV = \sum_{elements} f_j \int_{V_{element}} W_i N_j dV \qquad (2.6)$$

Clearly the trial and weighting functions will be known *a priori* and therefore their integrals can be evaluated. Consequently, given values for f at all discrete points we have a system of simultaneous algebraic equations that can be solved for values u^*_j throughout the domain.

The method outlined above forms the basis of all finite element techniques. Frequently, the weighting functions are chosen such that W_i=N_i, this is known as the Galerkin method. Other choices of weighting function have been suggested such as the Petrov-Galerkin and Spectral methods and these offer advantages over the Galerkin method in certain circumstances. However, the Galerkin method still remains the most frequently used technique for a wide variety of problems.

Returning now to the finite volume method consider again equation (2.1) integrated over a volume V.

$$\int_V \left(\nabla^2 u - f\right) dV = 0 \qquad (2.7)$$

Applying the divergence theorem we have

$$\int_A \nabla u.\mathbf{n} dA = \int_V f dV \qquad (2.8)$$

where A is the surface enclosing V and $\mathbf{n}$ is normal to that surface. Now assuming that V is small i.e. ΔV, we can make the following approximation

$$\sum_{faces} \left(\nabla u.\mathbf{A}\right)_{face} = f\Delta V \qquad (2.9)$$

where the summation on the left hand side is over the faces of ΔV, typically a cube or a tetrahedron, the subscript *face* indicates evaluation at that face and f is assumed to be a known constant over ΔV. Clearly, all that is required to turn equation (2.9) into a discrete relationship, is to express $(\nabla u.\mathbf{A})_{face}$ in terms of discrete points in the domain. Typically the discrete points are taken as the volume centres or the volume vertices of the grid covering the domain of interest. With a discrete expression for $(\nabla u.\mathbf{A})_{face}$ defined, then equation (2.9) can be coupled with similar relationships for the other volumes in the domain of interest, and then solved simultaneously for approximate values of u.

The prevalence of the finite element method in unstructured calculations is undoubtedly because the discretisation strategy itself leads to the use of triangular elements in two dimensions and tetrahedrons in three dimension. These elements are invariably the components that form an unstructured grid and hence it is natural to use this form of grid with all it's flexibility. However, the finite volume method has the important advantage that it is easier to ensure that mass conservation is satisfied throughout a solution domain, and because of this has become the method of choice in most CFD calculations.

2.3 SIMPLE Based Solution Algorithms

Coupled with the finite volume discretisation technique, the Semi Implicit Method for Pressure Linked Equations[15] (SIMPLE) and a number of it's derivatives have become some of the most popular methods for solving the incompressible Navier-Stokes equations. A number of these methods will be reviewed here however before that, it is necessary to discuss a problem that all these algorithms suffer from, velocity and pressure decoupling.

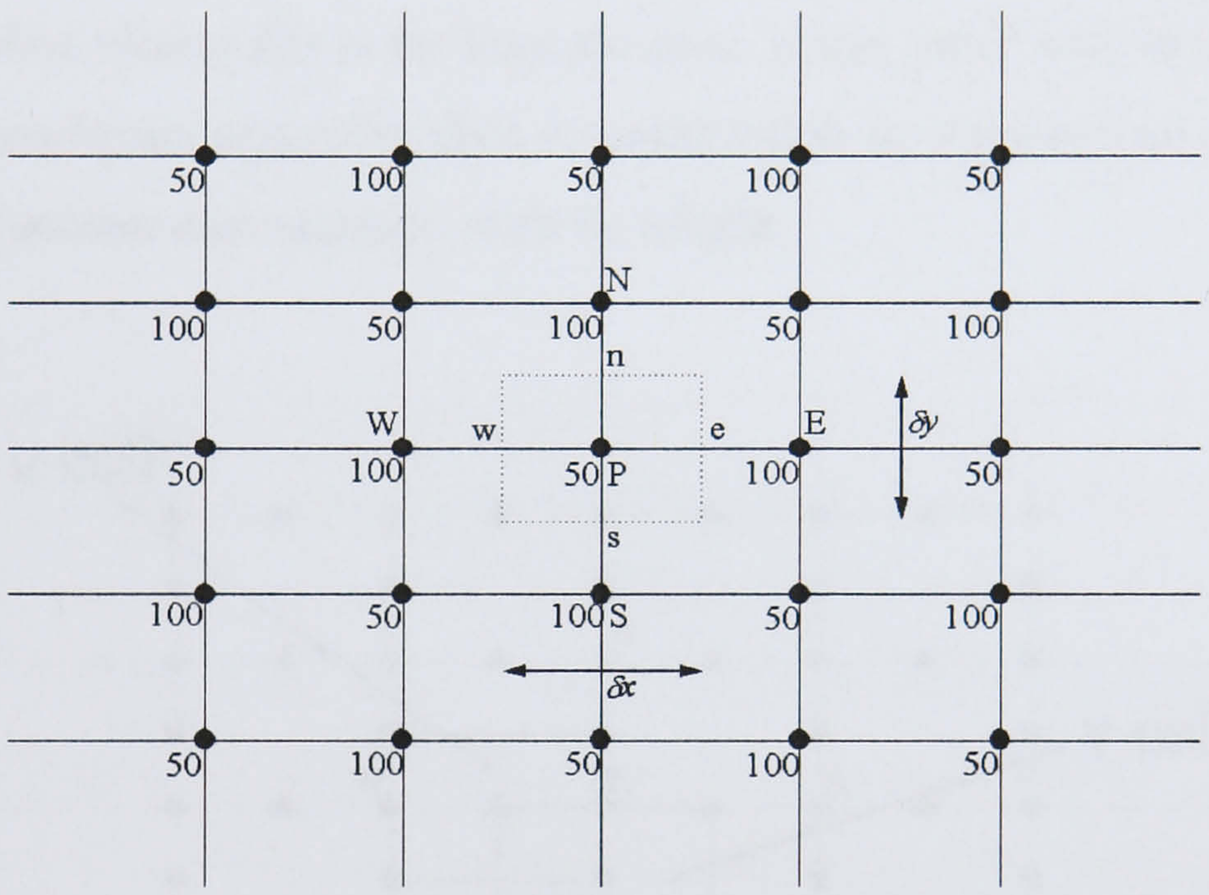


Figure 2.1: "Checkerboard" Pressure Field

A finite volume discretisation begins by splitting the flow domain into a set of discrete volumes. Flow variables are then stored at the centre of these volumes, or at their vertices, and the governing equations expressed in terms of algebraic relationships linking these discrete points. It would seem logical to store all flow variables, i.e. velocity, pressure and turbulent quantities, at the same locations throughout the computational domain. However, if velocities and pressures are stored at the same locations a highly non-uniform field pressure field can act like a uniform field in the discretised momentum equations. To demonstrate this, consider the grid and "checker-board" pressure field shown in Figure 2.1. If pressures at the *e* and *w* locations are obtained through linear interpolation then in two dimensions the pressure gradient term at *P* will be given by

$$\nabla p = \begin{pmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{p_e - p_w}{\delta x} \\ \frac{p_n - p_s}{\delta y} \end{pmatrix} = \begin{pmatrix} \frac{\left(\frac{p_E + p_P}{2}\right) - \left(\frac{p_P + p_W}{2}\right)}{\delta x} \\ \frac{\left(\frac{p_N + p_P}{2}\right) - \left(\frac{p_P + p_S}{2}\right)}{\delta y} \end{pmatrix} = \begin{pmatrix} \frac{p_E - p_W}{2\delta x} \\ \frac{p_N - p_S}{2\delta y} \end{pmatrix} \qquad (2.10)$$

Obviously, the central node P does not appear in the final form of equation (2.10). Therefore, if the appropriate values of the "checker-board" pressure field are substituted into equation (2.10), the resulting discretised gradient is zero throughout the domain when clearly this is far from the case. If this result were to be used in the discretised momentum equations then it would result in non-physical solutions and consequently another discretisation must be sought.

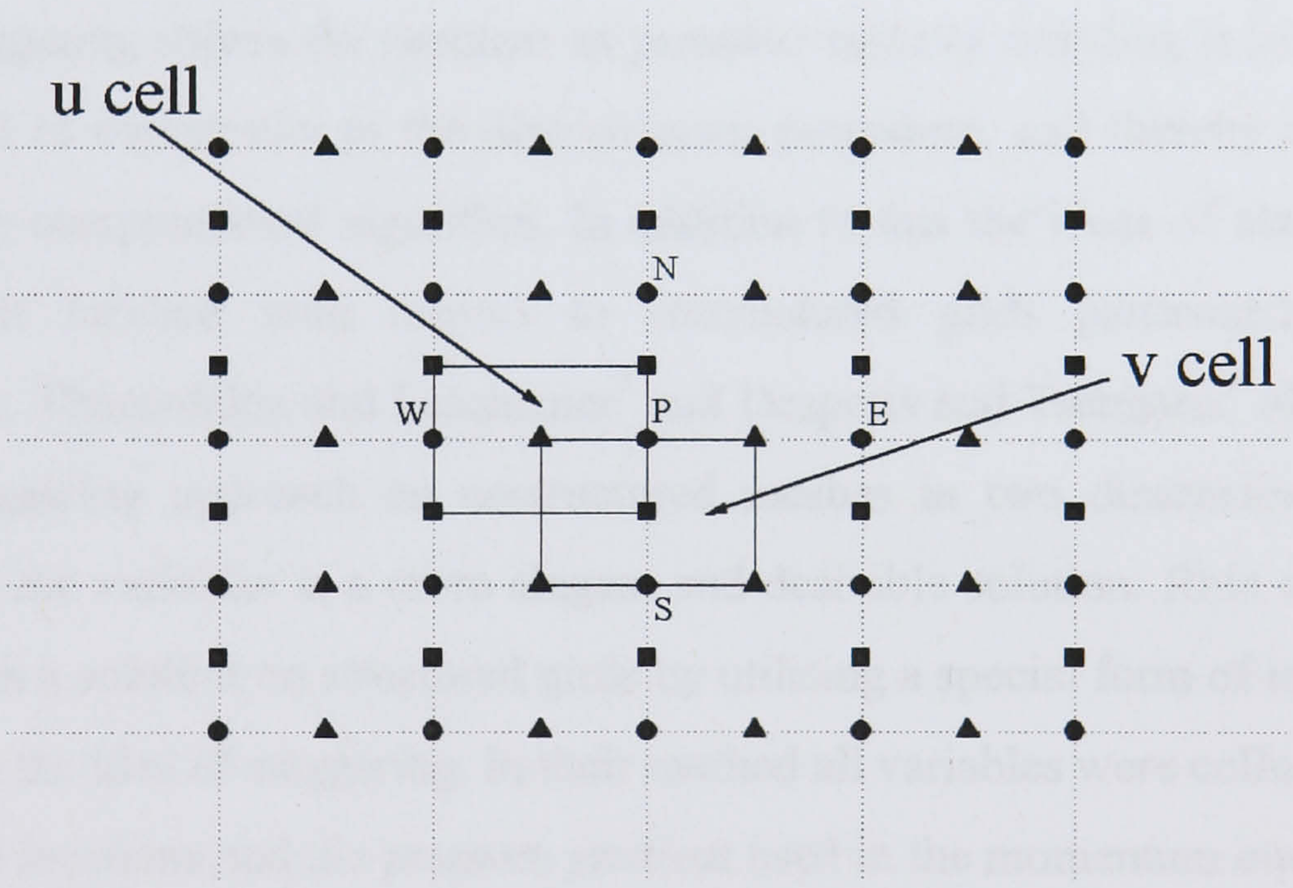


Figure 2.2: Schematic of the Staggered Grid. Triangle, Square and Circle Denote u, v and Pressure Storage Respectively

Harlow and Welch[14] suggested a remedy for the problem of pressure velocity decoupling, the staggered grid. In this form of grid the pressure and any other scalar variables are stored at the ordinary nodal locations shown in Figure 2.1 whilst the velocity components are staggered around these centres. A two-dimensional schematic of the arrangement is shown in Figure 2.2. We observe that the control volumes used for the two velocity components are different from each other and

different from the scalar control volumes. This enables the pressure gradient for the x momentum equation to be evaluated using the following expression

$$\frac{\partial p}{\partial x} = \frac{p_P - p_W}{\delta x} \qquad (2.11)$$

and similarly for the y momentum equation we have

$$\frac{\partial p}{\partial y} = \frac{p_P - p_S}{\delta y} \qquad (2.12)$$

If we consider the "checker-board" pressure field again then substitution of the appropriate values into equations (2.11) and (2.12) yields significant non-zero pressure gradients as would be expected.

Whilst staggering solves the problem of pressure velocity coupling it introduces an added level of complexity to the discretisation procedure, and thereby complicates the ensuing computational algorithm. In addition to this the ideas of staggering are not straight forward with respect to unstructured grids particularly in three dimensions. Thomadakis and Leschziner[7] and Despotis and Tsangaris[7] all attempted a semi-staggering approach on unstructured meshes in two dimensions however collocating the variables is a more elegant and desirable solution. Rhie and Chow[20] offered such a solution on structured girds by utilising a special form of interpolation based upon the idea of staggering. In their method all variables were collocated at the same nodal locations and the pressure gradient used in the momentum equations was identical to equation (2.10). However when the continuity equation is solved, equation (2.10) is replaced by equations (2.11) and (2.12) where appropriate and thus pressure and velocity decoupling is avoided. A full derivation of this method for unstructured meshes is presented in Chapter 3 and therefore more details are not given here.

2.3.1 The SIMPLE Algorithm

The difficulty in solving the incompressible Navier-Stokes equations is that there is no obvious equation to solve for the pressure field. The SIMPLE algorithm[15] overcomes this problem through the use of a guess and then correct procedure for the

pressure field. In other words, a pressure field is guessed, the momentum equations solved and then the pressure field updated through an appropriate mechanism. The mechanism used is the continuity equation because given a correct pressure field a solution of the momentum equations will also satisfy the continuity equation. This fact is used in the SIMPLE algorithm by casting the continuity equation in the form of a pressure correction equation. This can then be solved and used to update the pressure field and velocities used in the solution of the momentum equations. Again a detailed derivation of this pressure correction equation is left until Chapter 3, nevertheless for comparison with similar methods it is useful to present the steps involved in the SIMPLE algorithm below

1. Guess an initial pressure field
2. Solve the momentum equations
3. Solve the pressure correction equation
4. Correct the pressure and velocities
5. Solve all other discretised scalar transport equations
6. If the solution has converged stop, if not return to step 2

2.3.2 The SIMPLER Algorithm

Patankar[15] introduced the SIMPLE Revised (SIMPLER) algorithm in an attempt to increase the convergence rate of the standard SIMPLE algorithm. In this algorithm rather than deriving a pressure correction equation, pseudo-velocities are introduced in order to find a discretised pressure equation. Thus the intermediate pressure field is found directly without the use of a correction which is advantageous as the pressure correction often does a poor job of correcting the pressure field. However, the pressure correction is still found because it performs well at correcting velocities. The steps in the SIMPLER algorithm are

1. Calculate pseudo-velocities
2. Solve the pressure equation

3. Solve the discretised momentum equations

4. Solve the pressure correction equation

5. Correct velocities

6. Solve all other discretised scalar transport equations

7. If the solution has converged stop, if not return to step 2

2.3.3 The SIMPLEC Algorithm

The SIMPLE Consistent (SIMPLEC) algorithm which was developed by Van Doormal and Raithby[21] is identical to the SIMPLE algorithm with one exception. This is that the momentum equations are manipulated so that the velocity correction equations omit terms that are less significant than those omitted in SIMPLE. Consequently, the steps in the algorithm are identical to SIMPLE, see section 2.3.1.

2.3.4 The PISO Algorithm

Issa[22] adapted a pressure velocity calculation procedure, developed originally for the non-iterative computation of unsteady compressible flows, into an iterative solution for steady state incompressible problems. The method was called the Pressure Implicit with Splitting of Operators (PISO) algorithm and it involves one predictor step and two corrector steps and can be regarded as an extension of the SIMPLE algorithm. Essentially two pressure correction equations are formed in the PISO method the first as would be normally done in the SIMPLE algorithm and the second using the corrected velocities from the first corrector step. The algorithm can be summarised as follows

1. Perform steps 1 to 3 of the SIMPLE algorithm

2. Solve the second pressure correction equation

3. Correct the pressure and velocities

4. Solve all other discretised scalar transport equations

5. If the solution has converged stop, if not return to step 2 of the SIMPLE algorithm

The SIMPLE algorithm is fairly straightforward and has been implemented in numerous CFD codes. The other variations of SIMPLE can produce savings in computational effort due to improved convergence in certain situations. However, comparisons have shown that the performance of each algorithm depends on the flow conditions, the degree of coupling between the momentum equation and scalar equations and on the amount of underrelaxation used. Given all these different influences it is difficult to categorically state that any one of the algorithms is better than another and therefore unfortunately a particular algorithm must be chosen on a problem by problem basis. A comprehensive comparison of SIMPLER, SIMPLEC and PISO is given by Jang *et al.*[23] and this can be used to make an informed judgement on which algorithm is suitable for a given application.

2.4 Convective Modelling

To facilitate a review of convective modelling consider the following differential equation

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x}\left(-u\phi + \Gamma \frac{\partial \phi}{\partial x}\right) + S \qquad (2.13)$$

Equation (2.13) is the unsteady one-dimensional convection-diffusion equation for a scalar ϕ. It governs a number of different flow problems including the conservation of momentum in a Newtonian fluid where ϕ is taken as the fluid's velocity multiplied by it's density, Γ the kinematic viscosity and S the pressure gradient. In multi-dimensional problems equation (2.13) can also represent the unsteady transport of ϕ, if the extra components of convection and diffusion are grouped into the source term S. Therefore, we can regard equation (2.13) as a general representation of the convection and diffusion of ϕ through a fluid, and as such it can be used without loss of generality to discuss various convective models.

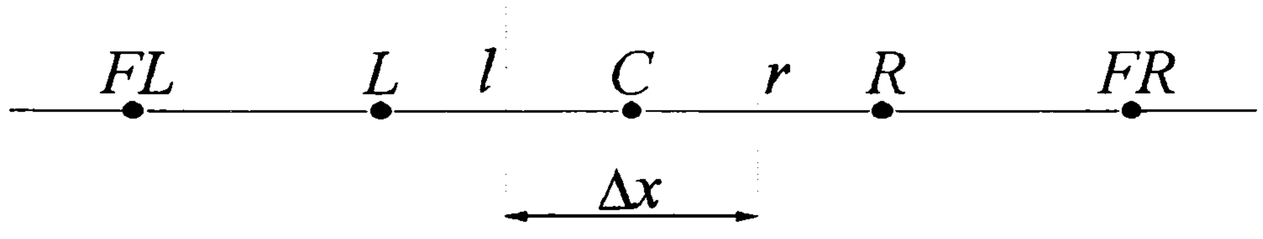


Figure 2.3: One-Dimensional Control Volume Schematic

Consider then the simple one-dimensional grid shown in Figure 2.3. Adopting the finite volume approach to discretisation, discussed earlier in this chapter, equation (2.13) is integrated over the control volume surrounding node *C* shown in Figure 2.3 i.e.

$$\int_l^r \frac{\partial \phi}{\partial t} dx = \int_l^r \frac{\partial}{\partial x}\left(-u\phi + \Gamma \frac{\partial \phi}{\partial x} \right) dx + \int_l^r S dx \qquad (2.14)$$

Equation (2.14) can now be approximated by the following expression

$$\overline{\frac{\partial \phi}{\partial t}} = \frac{1}{\Delta x}\left[(u\phi)_l - (u\phi)_r + \left(\Gamma \frac{\partial \phi}{\partial x} \right)_r - \left(\Gamma \frac{\partial \phi}{\partial x} \right)_l \right] + \bar{S} \qquad (2.15)$$

where Δx is defined in Figure 2.3, the overbar indicates an average value for the control volume and the subscripts l and r denote evaluation of the relevant expression at the points shown in Figure 2.3. Discretising the diffusion term using a standard central differencing procedure equation (2.15) becomes

$$\overline{\frac{\partial \phi}{\partial t}} = \frac{1}{\Delta x}\left[(u\phi)_l - (u\phi)_r + \Gamma\left(\frac{\phi_R - 2\phi_C + \phi_L}{\Delta x} \right) \right] + \bar{S} \qquad (2.16)$$

where the nodes *R* and *L* are shown in Figure 2.3 and a subscript again denotes evaluation of the appropriate expression at a particular point. The unsteady convection-diffusion is now in a general discretised form that can be used to implement and analyse various convective models.

2.4.1 First Order Upwinding and Central Differencing

The first two convective models to be considered here, first order upwinding and central differencing, are perhaps the most simple conceptually and also in terms of

implementation. However, they both have inherent flaws i.e. instability and inaccuracy that will be highlighted in this section.

Leonard[17] showed that the stability of a convective model may be defined as the sensitivity of the convective term to variations in ϕ_C. Consider then the situation, during the solution of equation (2.16), in which the convective term becomes too large to balance this equation, ϕ_C will then falsely increase over computational time as,

$$\frac{\partial \phi_C}{\partial t} \propto \overline{\frac{\partial \phi}{\partial t}} \qquad (2.17)$$

How this increase in ϕ_C will affect the convective term is dependent on the following expression

$$\Sigma_{CON} = \frac{\partial(\text{INFLUX})}{\partial \phi_C} \begin{cases} < 0 \text{ stable sensitivity} \\ = 0 \text{ neutral sensitivity} \\ > 0 \text{ unstable sensitivity} \end{cases} \qquad (2.18)$$

where Σ_{CON} is the convective sensitivity and INFLUX is the net convective influx into the control volume centred at *C*. Obviously, if ϕ_C has been falsely increased then $\Sigma_{CON}<0$ will decrease the convective term producing a stabilising effect over time whereas $\Sigma_{CON}>0$ will increase the convective term leading to further false increases in ϕ_C and an unstable solution. An analogous expression for diffusive sensitivity, Σ_{DIFF}, can be calculated and these two expressions combine to give an overall measure of the stability of a particular scheme. It should be noted that iterative solutions of steady state problems are governed by the same affects, in this case the iteration number is substituted for computational time.

Consider now the central differencing convective model. For a face *l* in Figure 2.3 this can be expressed as,

$$\phi_l = \frac{1}{2}\left(\phi_L + \phi_R\right) \qquad (2.19)$$

Substituting equation (2.19) and a similar expression for ϕ_r into equation (2.16) we have,

$$\overline{\frac{\partial \phi}{\partial t}} = \frac{1}{\Delta x}\left[\frac{u_l}{2}(\phi_L + \phi_C) - \frac{u_r}{2}(\phi_R + \phi_C) + \Gamma\left(\frac{\phi_R - 2\phi_C + \phi_L}{\Delta x}\right)\right] + \overline{S} \qquad (2.20)$$

Calculating the convective and diffusive sensitivities gives

$$\Sigma_{CON}(\text{CENTRAL}) = \frac{u_l - u_r}{2\Delta x} \approx -\frac{1}{2}\frac{\partial u}{\partial x} \qquad (2.21)$$

and

$$\Sigma_{DIFF}(\text{CENTRAL}) = -\frac{2\Gamma}{\Delta x^2} \qquad (2.22)$$

Clearly Σ_{DIFF} is always negative and therefore has a stabilising influence on the solution. However, the influence of Σ_{DIFF} is related to the local Péclet number, P_Δ, defined as

$$P_\Delta = \frac{u_C \Delta x}{\Gamma} \qquad (2.23)$$

If P_Δ is large then Σ_{DIFF} will be very small and therefore have a limited stabilizing affect. Σ_{CON}'s value is dependent on the local flow velocity gradient and in the presence of flow deceleration will be positive creating an unstable influence on the solution. Therefore, in the case of large local Péclet number, P_Δ>2, Σ_{DIFF} may be too small to stabilise the solution in areas where flows decelerate. In general it is difficult to ensure a stable sensitivity if central differencing is used as the convective model and therefore makes it a poor choice of model.

Consider now the first order upwind scheme for face *l*,

$$(u\phi)_l = \langle u_l, 0\rangle \phi_L - \langle -u_l, 0\rangle \phi_C \qquad (2.24)$$

where ⟨A,B⟩ denotes the greater of A and B. Here, face *l* is given the same value as the node immediately upstream of it. This upstream bias is based on the physical situation being modelled. As convection is essentially the transport of fluid

properties from upstream to downstream any numerical model should represent this characteristic. Substituting equation (2.24) and an analogous expression for face *r* into equation (2.16) gives,

$$\overline{\frac{\partial \phi}{\partial t}} = \frac{1}{\Delta x}\left(\langle u_l,0\rangle \phi_L - \langle -u_l,0\rangle \phi_C - \langle u_r,0\rangle \phi_C + \langle -u_r,0\rangle \phi_R\right) \qquad (2.25)$$
$$+\Gamma\left(\frac{\phi_R - 2\phi_C + \phi_L}{\Delta x^2}\right) + \overline{S}$$

Calculating Σ_{CON} for the upwind scheme we have,

$$\Sigma_{CON}(\text{UPWIND}) = -\frac{1}{\Delta x}\left[\langle u_r,0\rangle + \langle -u_l,0\rangle\right] \qquad (2.26)$$

The four combinations of signs of u_l and u_r produce three stable and one neutral sensitivity. In contrast to the central differencing case this yields a very stable convective model. However, closer examination of the discretised equation shows that stability is being gained by the introduction of numerical diffusion to the solution. Consider the situation when the fluid has a uniform velocity u_0 in the positive *x* direction. Equation (2.25) then becomes,

$$\overline{\frac{\partial \phi}{\partial t}} = u_0 \frac{\phi_L - \phi_C}{\Delta x} + \Gamma\left(\frac{\phi_R - 2\phi_C + \phi_L}{\Delta x^2}\right) + \overline{S} \qquad (2.27)$$

Equation (2.27) can be rewritten in the following form,

$$\overline{\frac{\partial \phi}{\partial t}} = u_0 \frac{\phi_L - \phi_R}{2\Delta x} + \left(\Gamma + \Gamma_{num}\right)\left(\frac{\phi_R - 2\phi_C + \phi_L}{\Delta x^2}\right) + \overline{S} \qquad (2.28)$$

where,

$$\Gamma_{num} = \frac{u_0 \Delta x}{2} \qquad (2.29)$$

The first term in equation (2.28) is the uniform velocity form of the central differencing model given by equation (2.20). The second term is the normal central differencing diffusion term with an added numerical diffusion Γ_{num}. So, the first

order upwind model is in fact a form of central differencing with an artificial diffusion term added to ensure stability. In quasi-one-dimensional flow the introduction of Γ_{num} does not present too great a problem. The extra diffusion is in the direction of flow and consequently it will be dominated by the convective term and therefore have little effect. But, in multi-dimensional flow an artificial cross-wind[24] diffusion is created that can lead to extremely inaccurate results. In these circumstances it essential to ensure that artificial diffusion is insignificant. This can be guaranteed by enforcing the following condition on the local Péclet number,

$$P_\Delta = \frac{u_0 \Delta x}{\Gamma} << 2 \qquad (2.30)$$

Condition (2.30) like the restriction to low local Péclet numbers in the central differencing model is often impossible to enforce. For example, P_Δ is equivalent to local Reynolds number in the momentum equation form of equation (2.13) and engineering problems of practical interest often have much larger global Reynolds numbers than two. Consequently, to achieve local Reynolds numbers which satisfy condition (2.30) grid size would have to be so small that the resulting system of linear equations would be prohibitively large to solve in a reasonable time frame.

In addition to introducing numerical diffusion into the solution upwinding also suffers from a low order of accuracy. In principle the accuracy can always be improved by reducing grid size but analysis shows that this is not necessarily the case. Consider equation (2.27) rewritten in the following form,

$$u_0 \frac{\phi_C - \phi_L}{\Delta x} = \bar{S}', \qquad (2.31)$$

the unsteady and diffusive term have been absorbed into the source term leaving just the convective term to be evaluated explicitly. The corresponding differential equation is given by,

$$u_0 \frac{\partial \phi}{\partial x} = S' \qquad (2.32)$$

Assuming an exact solution ϕ^E exists then $\bar{S}'$ can be evaluated by integrating equation (2.32) between l and r,

$$\bar{S}' = \frac{1}{\Delta x}\int_l^r u_0 \frac{\partial \phi}{\partial x} dx = u_0 \frac{\left(\phi_r^E - \phi_l^E\right)}{\Delta x} \qquad (2.33)$$

Substituting equation (2.33) into equation (2.31) we have,

$$\phi_C - \phi_L = \phi_r^E - \phi_l^E \qquad (2.34)$$

Using the exact solution or computed results an estimate of the error ε can be made by constructing a graph using equation (2.34). Figure 2.4 shows an example of such a graph and it can be clearly seen that the error in the upwind solution is given by,

$$\varepsilon = \phi^E - \phi \approx \frac{\Delta x}{2}\left[\left(\frac{\partial \phi^E}{\partial x}\right)_{\text{ref}} - \left(\frac{\partial \phi^E}{\partial x}\right)\right] = \frac{1}{2}\left(\delta\phi_{\text{ref}}^E - \delta\phi^E\right) \qquad (2.35)$$

where the derivatives $\partial\phi^E/\partial x$ and $(\partial\phi^E/\partial x)_{\text{ref}}$ are the local slope of the exact solution and the slope at the corresponding common boundary reference point respectively, and the $\delta\phi^E$'s are the analogous increases in ϕ^E across a control volume shown in Figure 2.5.

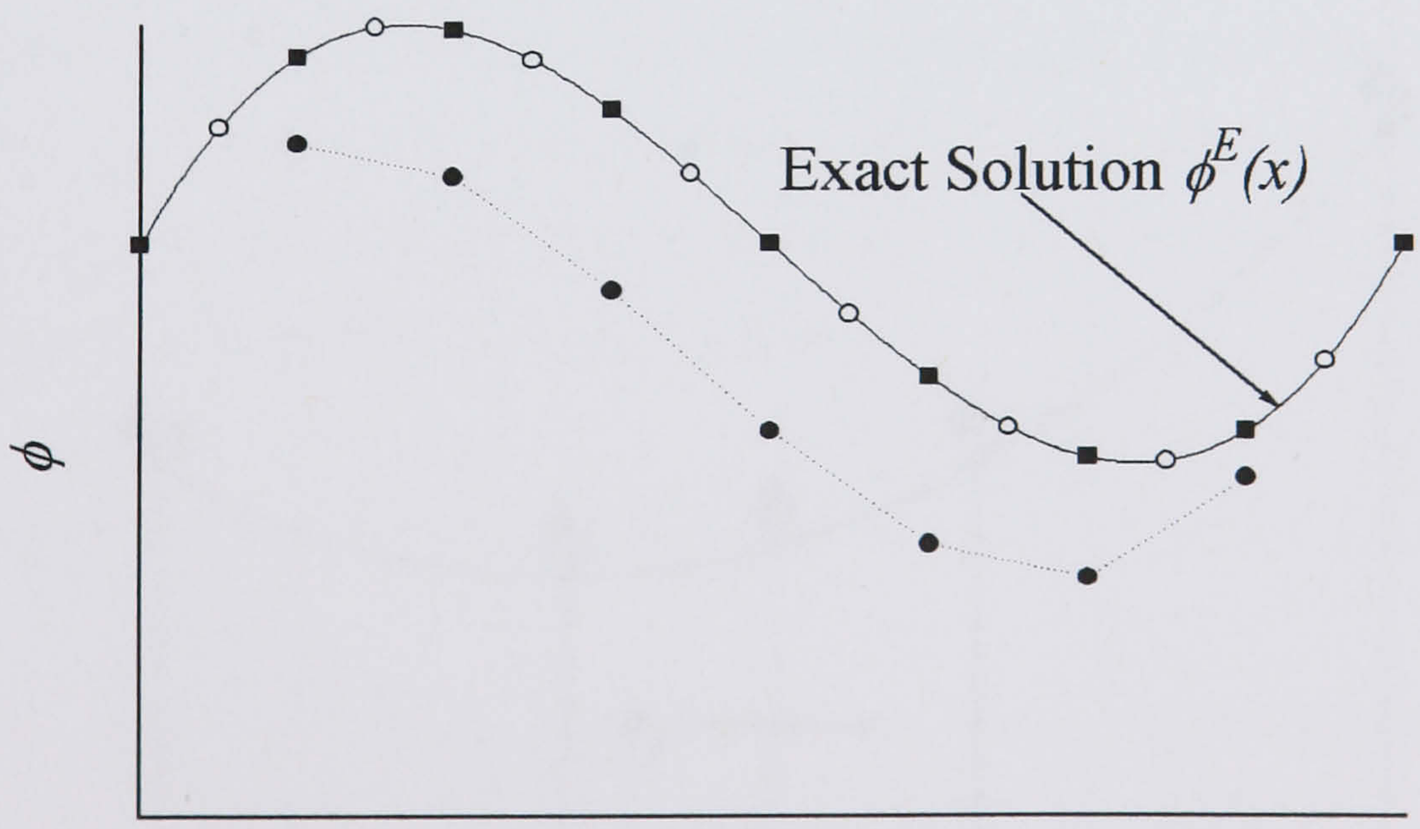


Figure 2.4: Graphical Construction of the First Order Upwind Solution Given the Exact Solution

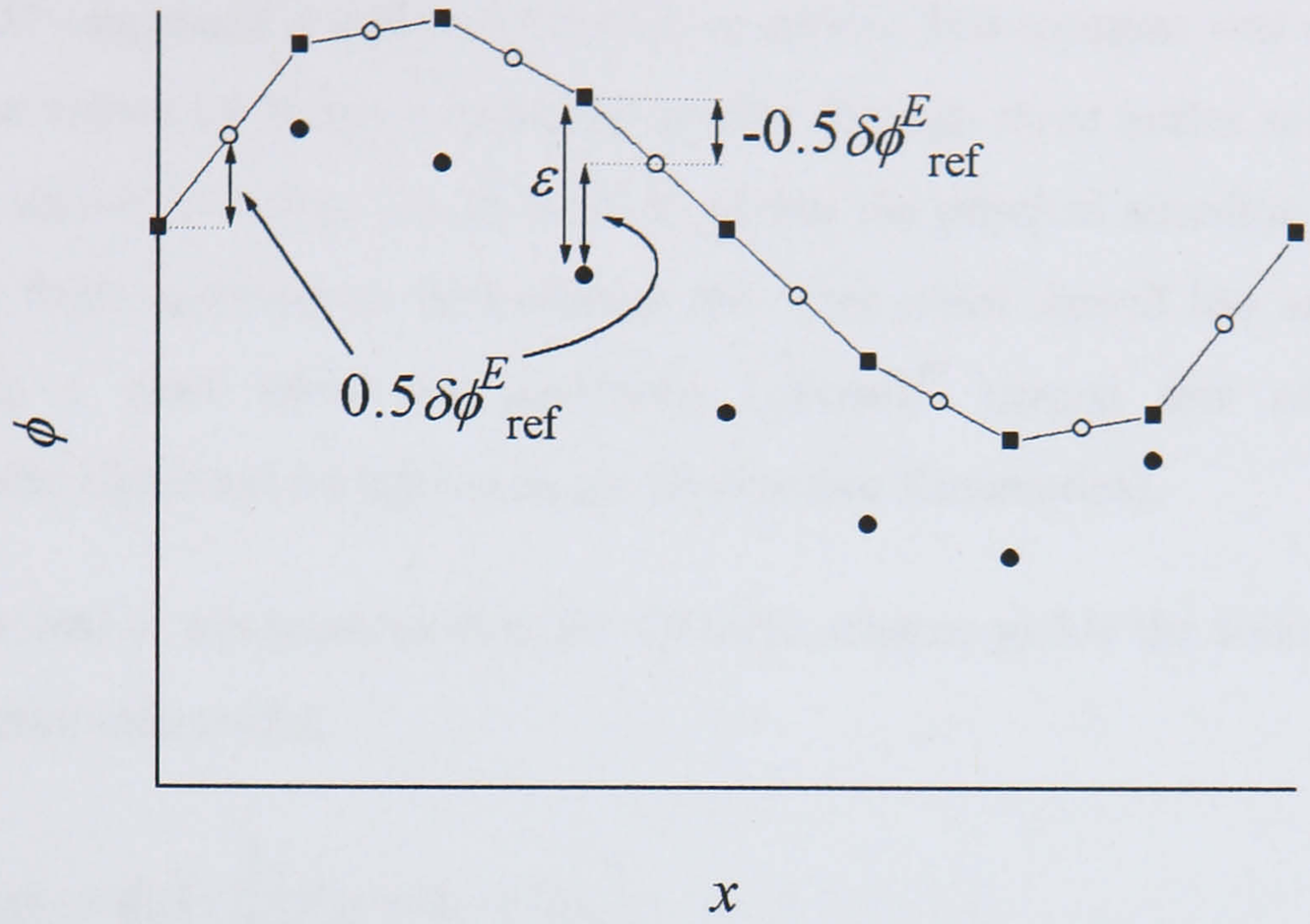


Figure 2.5: An Estimate of Error Using the Cell Increments

Equation (2.35) demonstrates that although ε is proportional to the grid size Δx it is also dependent on variations in ϕ. If the gradients of ϕ span a large of range values then the error may be considerably higher than first order. This problem and the restriction that $P_{\Delta}<<2$ constrains the upwind scheme to accurately modelling only a limited number of flow situations and therefore limits it's practical use.

2.4.2 The QUICK Scheme

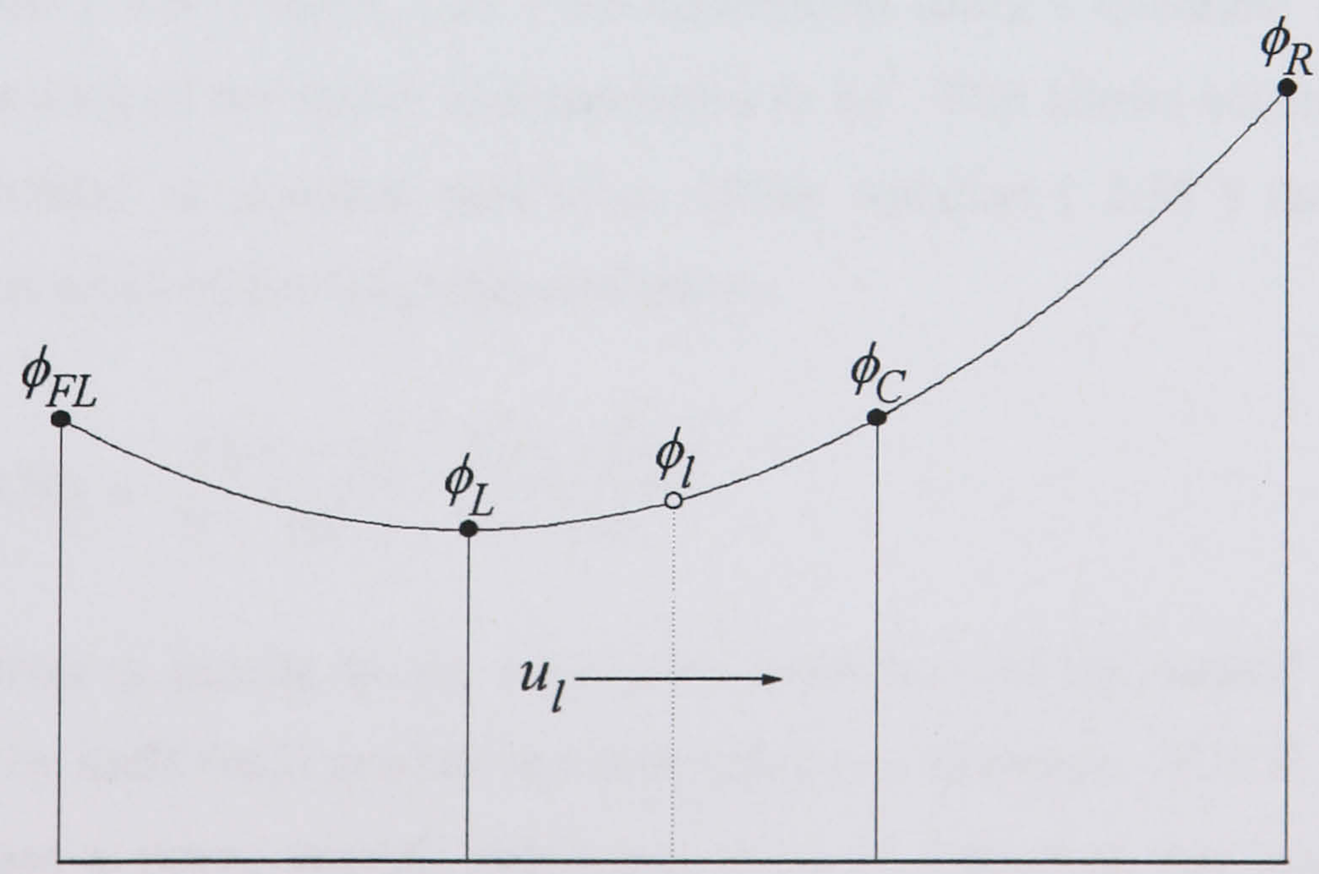


Figure 2.6: Three Point Stencil for the QUICK Model

To avoid the instability of central differencing and the inaccuracy of upwinding Leonard[17] suggested a different convective model. His solution was to calculate the cell face values by fitting a quadratic profile through three nodes surrounding each face, as shown in Figure 2.6. In order to mirror the physical situation of information moving from upstream to downstream the three point stencil has an upwind bias. Showing a great talent for acronyms Leonard[17] named this scheme QUICK (Quadratic Upstream Interpolation for Convective Kinematics).

When u_l and u_r are positive then the QUICK scheme yields the following formulae for the face values of ϕ,

$$\phi_l = \frac{1}{2}\left(\phi_L + \phi_C\right) - \frac{1}{8}\left(\phi_{FL} + \phi_C - 2\phi_L\right) \qquad (2.36)$$

$$\phi_r = \frac{1}{2}\left(\phi_C + \phi_R\right) - \frac{1}{8}\left(\phi_L + \phi_R - 2\phi_C\right)$$

Similarly when u_l and u_r are negative we have,

$$\phi_l = \frac{1}{2}\left(\phi_C + \phi_L\right) - \frac{1}{8}\left(\phi_R + \phi_L - 2\phi_C\right) \qquad (2.37)$$

$$\phi_r = \frac{1}{2}\left(\phi_R + \phi_C\right) - \frac{1}{8}\left(\phi_{FR} + \phi_C - 2\phi_R\right)$$

As equations (2.36) and (2.37) are constructed using a quadratic interpolation scheme the error of the model is proportional to Δx^3. This allows accurate solutions to be calculated on practical grid sizes. Using equation (2.36) the convective sensitivity is given by the following expression,

$$\Sigma_{CON}(\text{QUICK}) = -\frac{1}{2}\frac{\left(u_r - u_l\right)}{\Delta x} - \frac{1}{8}\frac{\left(u_l + 2u_r\right)}{\Delta x} \qquad (2.38)$$

The first term is similar to the convective sensitivity of the central differencing model and by itself could produce unstable solutions. However, QUICK has an extra term that has a strong enough stabilising effect to overcome this instability. The QUICK scheme would seem to solve the problems of inaccuracy and stability that

plagued the central differencing and upwind models, but it's practical implementation is not without difficulty.

Leonard[17] suggested that the first term in equations (2.36) and (2.37) be evaluated implicitly and the second term treated as an explicit source term. In other words the convective term is modelled using central differencing and a correction is introduced into the source term which is lagged by one iteration. A number of papers in the open literature, Leschziner[25], Han *et al.*[26] and Pollard and Siu[27], tested this form of the QUICK scheme and found that it did not always produce stable or bounded results. A number of fixes were offered to guarantee stability including the addition of a pseudo-source term[25] and the development of the QUICKER[26] (QUICK Extended and Revised form) model. However, none of these were a rigorous derivation of the QUICK scheme, which ensured convergence.

Nevertheless, Hayase *et al.*[28] analysed the QUICK scheme using the four rules given by Patankar[15] that ensure the stable convergence of a finite volume based algorithm towards a physically realistic solution. Invoking these rules produced the following form of the model,

$$\phi_l = \phi_L + \frac{1}{8}\left(-\phi_{FL} - 2\phi_L + 3\phi_C\right) \qquad (2.39)$$

for $u > 0$

$$\phi_r = \phi_C + \frac{1}{8}\left(-\phi_L - 2\phi_C + 3\phi_R\right)$$

and,

$$\phi_l = \phi_C + \frac{1}{8}\left(3\phi_L - 2\phi_C - \phi_R\right) \qquad (2.40)$$

for $u < 0$

$$\phi_r = \phi_R + \frac{1}{8}\left(3\phi_C - 2\phi_R - \phi_{FR}\right)$$

Clearly, this formulation of QUICK is simply the sum of a simple upwind estimation of the face value plus a correcting term. The first term is evaluated implicitly and the second term treated as an explicit source term. Although not vastly different to Leonard's[17] original derivation this form of QUICK satisfies Patankar's[15] four rules and therefore makes it a more robust model. The problem of boundedness also found

in the tests of [24], [25], [26] and [27] lead to the development of high order composite models, some of which include QUICK as an integral part, these will be considered next.

2.4.3 High Order Composite Models

A convective model is regarded as bounded if the computed solution generated by the model is free of unphysical overshoots or undershoots. Typically, if a scheme is unbounded a step-like profile in ϕ, which should be sharp and monotonic, will be calculated with an overshoot at one or both ends. Often these effects are small and can be ignored but in some cases the inaccurate prediction of sharp gradients in ϕ can lead to divergent solutions.

Leschziner[25] noted that the QUICK scheme suffered from unboundedness and other high order upwind schemes such as CUI[29] (Cubic Upwind Interpolation) and second order upwinding[30] are troubled by the same problem. Although the increased accuracy of the high order models is desirable the unboundedness, and as a consequence instability in certain situations, is not. This lead Gaskell and Lau[31] to develop the convective boundedness criterion (CBC) which set out the conditions for boundedness and gave birth to a family of bounded, accurate and stable models.

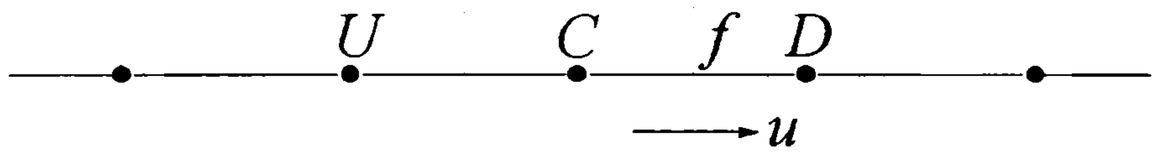


Figure 2.7: CBC Stencil of Nodes

The CBC is based on a normalised variable formulation (NVF). Consider face f of a control volume and the surrounding upstream (U), downstream (D), and central (C) nodes shown in Figure 2.7. As all the convective models discussed essentially have the same functional relationship that is,

$$\phi_f = f(\phi_U, \phi_C, \phi_D) \qquad (2.41)$$

it is useful to introduce a normalised variable defined as follows,

$$\tilde{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \qquad (2.42)$$

Using equation (2.42) the functional relationship is now simplified to,

$$\tilde{\phi}_f = f(\tilde{\phi}_C) \qquad (2.43)$$

as the normalised values of ϕ_D and ϕ_U are 1 and 0 respectively. This simplification is helpful in defining the conditions for boundedness and also allows the construction of a simple method for analysing stability and accuracy. The following table lists the models discussed so far in their original and normalised forms.

Model	Original Functional Relationship	Normalised Functional Relationship
Upwind	$\phi_f = \phi_C$	$\tilde{\phi}_f = \tilde{\phi}_C$
Central differencing	$\phi_f = \frac{1}{2}(\phi_C + \phi_D)$	$\tilde{\phi}_f = \frac{1}{2}(1 + \tilde{\phi}_C)$
QUICK	$\phi_f = \frac{1}{8}(3\phi_D + 6\phi_C - \phi_U)$	$\tilde{\phi}_f = \frac{3}{8} + \frac{3}{4}\tilde{\phi}_C$
CUI	$\phi_f = \frac{1}{6}(2\phi_D + 5\phi_C - \phi_U)$	$\tilde{\phi}_f = \frac{1}{3} + \frac{5}{6}\tilde{\phi}_C$
Second order upwinding	$\phi_f = \frac{1}{2}(3\phi_C - \phi_U)$	$\tilde{\phi}_f = \frac{3}{2}\tilde{\phi}_C$

Table 2.1: The Functional Relationships for a Number of Convective Models.

Using the NVF the CBC can be expressed as set conditions restricting the functional relationship given in equation (2.43). The conditions are defined as follows,

$$f(\tilde{\phi}_C) \text{ is continuous} \qquad (2.44)$$

$$f(0) = 0$$

$$f(1) = 1$$

$$f(\tilde{\phi}_C) < 1 \text{ and } f(\tilde{\phi}_C) > \tilde{\phi}_C \text{ for } 0 < \tilde{\phi}_C < 1$$

$$f(\tilde{\phi}_C) = \tilde{\phi}_C \text{ for } \tilde{\phi}_C < 0 \text{ or } \tilde{\phi}_C > 1$$

The criterion is illustrated diagrammatically in Figure 2.8.

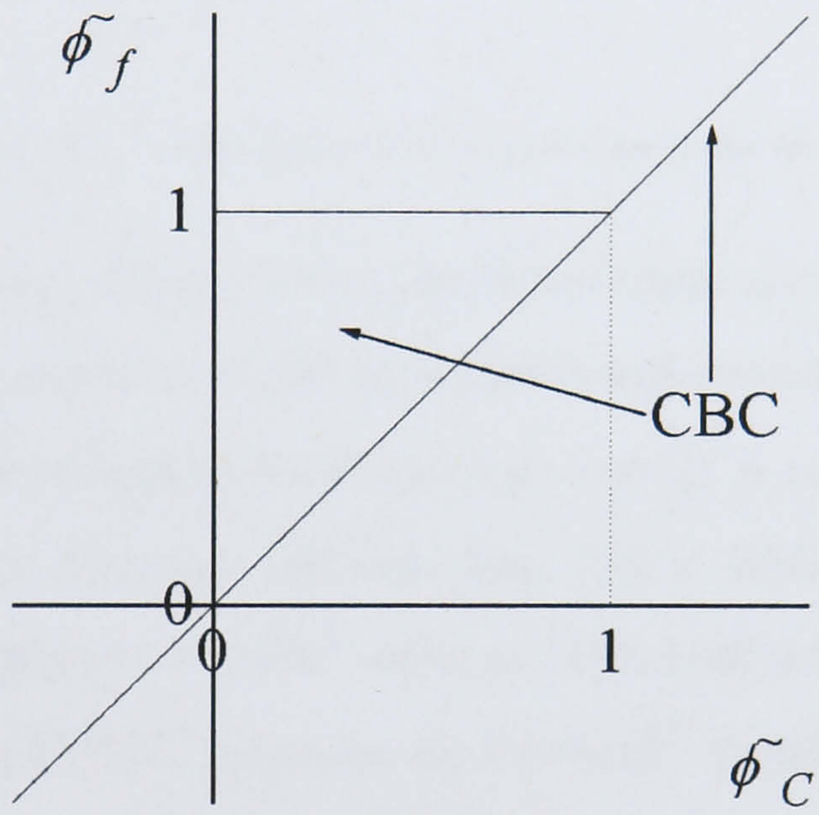


Figure 2.8: The Convective Boundedness Criterion

Clearly some of the CBC's conditions are very restrictive, in fact Figure 2.9 shows that the only convective model listed in Table 2.1 which satisfies the CBC in all regions is the upwind scheme. But, as has already been established this is achieved at the price of excessive numerical diffusion and inaccuracy. Also, it should be noted that the four other schemes plotted in Figure 2.9 pass through the point (0.5, 0.75). Leonard[32] has shown that if a models functional relationship passes through (0.5, 0.75) then it is a necessary and sufficient condition for second order accuracy. If in addition the function has a gradient of 0.75 when it passes through (0.5,0.75) then this is a necessary and sufficient condition for third order accuracy. The second of these conditions shows that only the QUICK scheme is third order accurate.

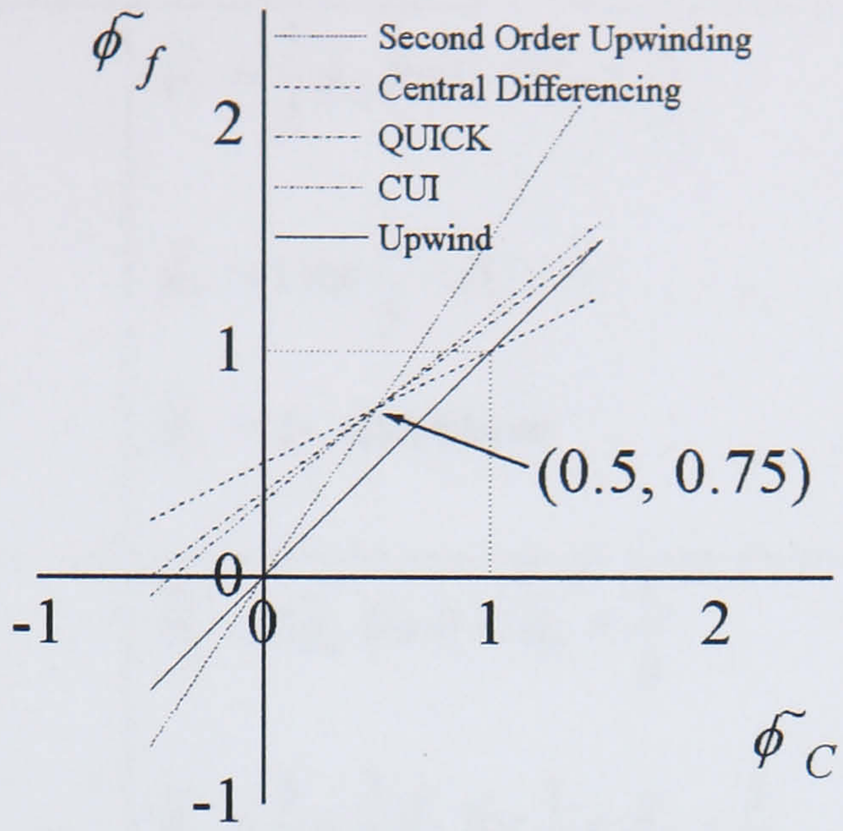


Figure 2.9: Various Normalised Linear Convective Models

It is evident from Figure 2.9 that a linear function cannot simultaneously be accurate and satisfy the CBC. In order to create an accurate and bounded scheme a non-linear function must relate the normalised values of ϕ_C and ϕ_f. A number of schemes have adopted this approach namely, Gaskell and Lau's SMART[31] scheme, Roe's MINMOD[33] scheme, Osher's OSHER[34] scheme, Van Leer's MUSCL[35] and CLAM[36] schemes and Darwish's STOIC[37] scheme. As Darwish[37] noted the MINMOD scheme is similar to Zhu and Rodi's SOUCOUP[38] scheme. Five of these schemes normalised functional relationships are presented in Table 2.2. Each scheme is a combination of linear functions with the exception of the CLAM model, which uses a quadratic relationship. The normalised functions of the five schemes are plotted in Figure 2.10. Clearly, this shows that all the schemes satisfy the CBC and have at least second order accuracy. The SMART and STOIC schemes achieve third order accuracy as at point (0.5, 0.75) both schemes have a gradient of 0.75.

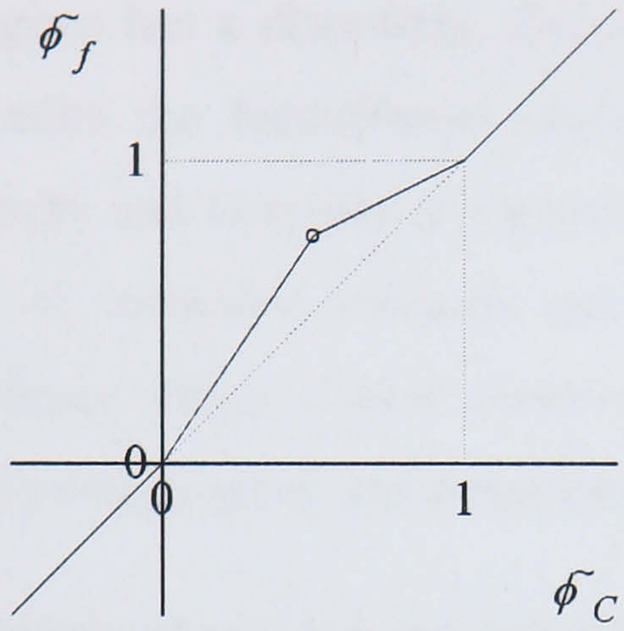
Model	Normalised functional Relationship
MINMOD or SOUCOUP	$\tilde{\phi}_f = \frac{3}{2}\tilde{\phi}_C$ for $0 < \tilde{\phi}_C < \frac{1}{2}$ $\tilde{\phi}_f = \frac{1}{2}\left(1 - \tilde{\phi}_C\right)$ for $\frac{1}{2} < \tilde{\phi}_C < 1$ $\tilde{\phi}_f = \phi_C$ elsewhere

OSHER	$\tilde{\phi}_f = \frac{3}{2}\tilde{\phi}_C$ for $0 < \tilde{\phi}_C < \frac{2}{3}$ $\tilde{\phi}_f = 1$ for $\frac{2}{3} < \tilde{\phi}_C < 1$ $\tilde{\phi}_f = \phi_C$ elsewhere
SMART	$\tilde{\phi}_f = 3\tilde{\phi}_C$ for $0 < \tilde{\phi}_C < \frac{1}{6}$ $\tilde{\phi}_f = \frac{3}{8} + \frac{3}{4}\tilde{\phi}_C$ for $\frac{1}{6} < \tilde{\phi}_C < \frac{5}{6}$ $\tilde{\phi}_f = 1$ for $\frac{5}{6} < \tilde{\phi}_C < 1$ $\tilde{\phi}_f = \phi_C$ elsewhere
STOIC	$\tilde{\phi}_f = 3\tilde{\phi}_C$ for $0 < \tilde{\phi}_C < \frac{1}{5}$ $\tilde{\phi}_f = \frac{1}{2}(1 + \tilde{\phi}_C)$ for $\frac{1}{5} < \tilde{\phi}_C < \frac{1}{2}$ $\tilde{\phi}_f = \frac{3}{8} + \frac{3}{4}\tilde{\phi}_C$ for $\frac{1}{2} < \tilde{\phi}_C < \frac{5}{6}$ $\tilde{\phi}_f = 1$ for $\frac{5}{6} < \tilde{\phi}_C < 1$ $\tilde{\phi}_f = \phi_C$ elsewhere
CLAM	$\tilde{\phi}_f = \tilde{\phi}_C(2 - \tilde{\phi}_C)$ for $0 < \tilde{\phi}_C < 1$ $\tilde{\phi}_f = \phi_C$ elsewhere

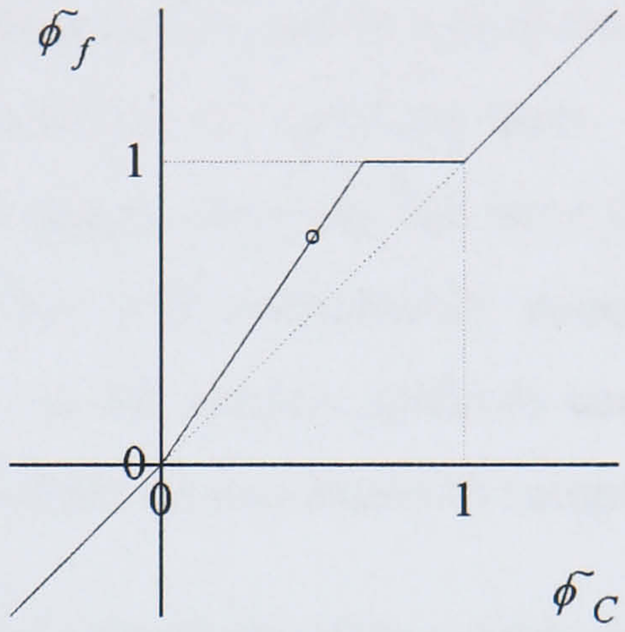
Table 2.2: The Normalised Functional Relationships for Five Schemes

Although these composite convective models have been widely used on structured orthogonal grids their application to unstructured meshes presents some significant difficulties. These difficulties are caused in the main by the lack of line structure in an unstructured mesh, which makes choosing a second upstream node for the three

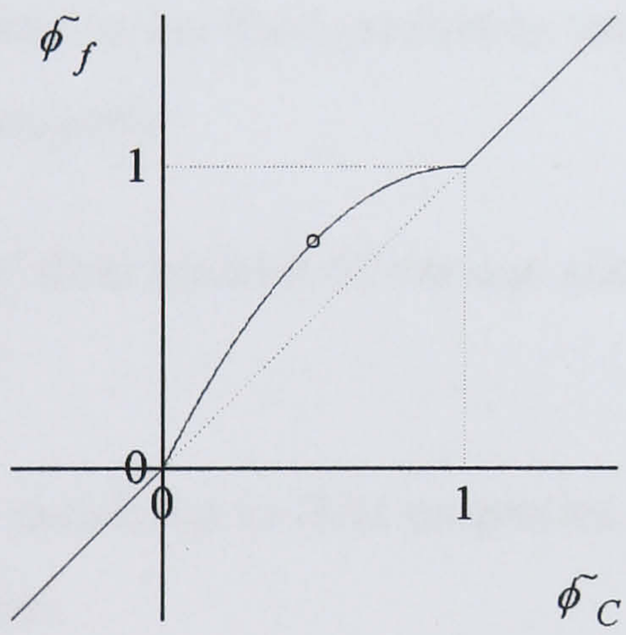
node stencil difficult. However, solutions are possible which incorporate all the advantageous features of the composite schemes presented here and they will be discussed later in this thesis.



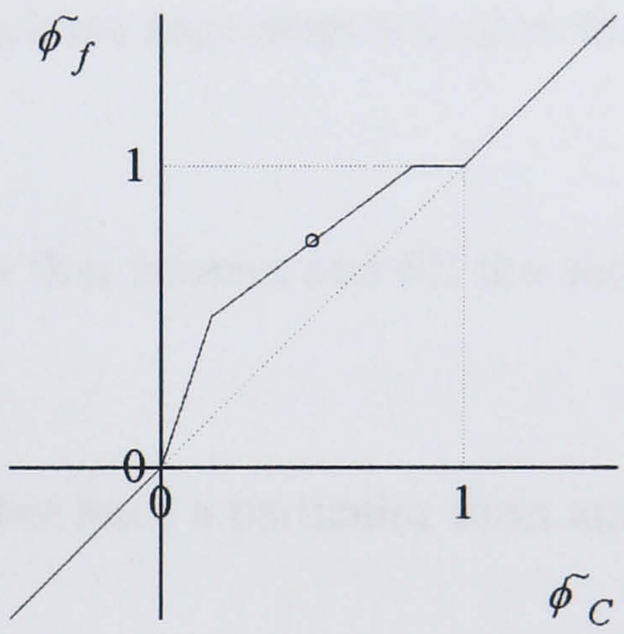
(a)



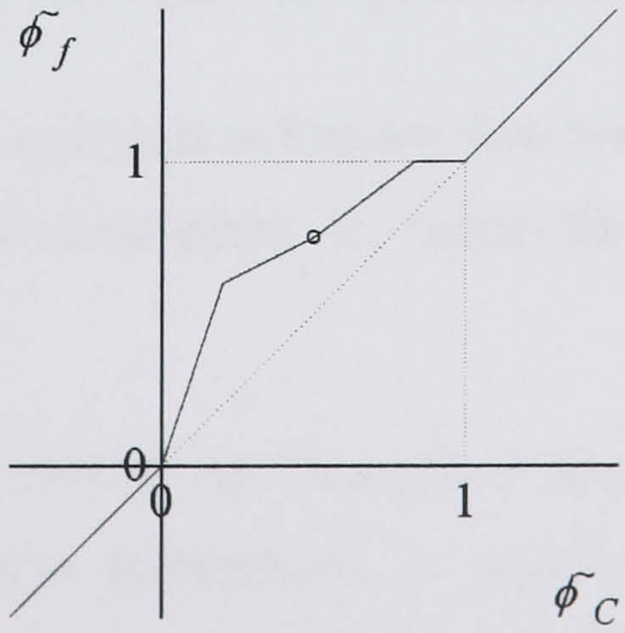
(b)



(c)



(d)



(e)

Figure 2.10: Normalised Plots of Five Composite Convective Models (a) MINMOD or SOUCOUP (b)OSHER (c) CLAM (d) SMART (e) STOIC

2.5 Turbulence Modelling

Once the Reynolds number of a given flow exceeds a critical value, laminar flow, characterized by it's stable, orderly behaviour will give way to a new flow regime. This new regime has a disorderly, fluctuating nature and is called turbulence. The ability to predict the fundamental characteristics of turbulent flow, i.e. velocity, pressure, density and viscosity at a given position and time, has been the subject of many years of extensive research and this will undoubtedly continue for the foreseeable future. Here, a brief overview of the various methods used to predict turbulent flow will be given and the merits of the various schemes compared.

To begin with consider a physical description of turbulent flow which characterises it using five key elements[39]. These are,

- Fluctuations in the fluid properties, which are superimposed upon the mean value of each property.

- Eddies of fluid packets of various sizes that interact and fill the shear layer near any wall

- Random variations in fluid properties that have a particular form and are not just white noise.

- Self-sustaining motion that once started can overcome the effects of viscous dissipation by producing new eddies to replace those lost.

- Mixing that is much stronger than in laminar flow because turbulent eddies move actively about in three dimensions to cause the rapid diffusion of mass, momentum and energy.

Whilst these characteristics give us the language to describe turbulent flow they do not give us any quantitative information or prediction ability. To do that a mathematical description of the flow is necessary and for that we turn to the Navier-Stokes equations.

As far as is known the Navier-Stokes equations are as valid for turbulent flows as they are for laminar flows. Therefore, solving Navier-Stokes for turbulent flow problems, using numerical methods, should provide one very direct method of predicting turbulent flow. However, the wide spectrum of flow scales involved in turbulent flow inevitably require that grid sizes become prohibitively large and therefore other solutions have been sought. Although, with the recent increases in computing power the area of Direct Numerical Simulation (DNS) has become a more active research area.

Since calculating the raw flow properties of turbulent flow in many cases is not practical, Reynolds[40] suggested another solution, time or Reynolds averaging. In this method each raw flow property is split into a time mean value plus a fluctuating value i.e.

$$u_i = U_i + u' \tag{2.45}$$

$$p = P + p' \tag{2.46}$$

where U_i and P are the time averaged components and u_i' and p' are the fluctuating components of velocity and pressure respectively. Substituting equations (2.45) and (2.46) into the incompressible Navier-Stokes and continuity equations and taking the time average of each equation individually yields the following relations

$$\frac{\partial}{\partial x_i}(\rho U_i) = 0 \tag{2.47}$$

$$\frac{\partial}{\partial x_j}\left(\rho U_i U_j - \mu\frac{\partial U_i}{\partial x_j} - \rho\overline{u_i' u_j'}\right) = -\frac{\partial P}{\partial x_i} \tag{2.48}$$

where the overbar indicates average.

Clearly, whilst the Reynolds averaged velocity components satisfy an identical continuity equation to the raw velocity components, the Reynolds averaged Navier Stokes (RANS) equations are complicated by a new term $\rho\overline{u_i' u_j'}$. This new term is called the Reynolds stress and introduces nine new variables into the system of

equations. As there are only four equations and a total of thirteen unknowns this presents a closure problem which can only be solved through the use of additional relationships or empirical modelling ideas.

2.5.1 Eddy Viscosity

The majority of methods to close the RANS equations in practical engineering applications are based upon the idea of eddy viscosity, first proposed by Boussinesq[41] in 1877. Essentially Boussinesq's hypothesis put forward the idea that the Reynold's stress could be modelled by a partially analogous expression to the viscous stress i.e.

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}\right)-\frac{2}{3}\rho k\delta_{ij} \qquad (2.49)$$

where U_i is the time mean velocity component, μ_t it the turbulent viscosity, δ_{ij} is the kronecker delta and k is the turbulent kinetic energy defined by

$$k=\frac{1}{2}\overline{u_i'u_i'} \qquad (2.50)$$

It is observed that μ_t has the same dimensions as viscosity, however it is not a fluid property, varying instead with flow conditions and position. Clearly, if an appropriate value for the turbulent viscosity can be found then substitution of equation (2.49) into equation (2.48) will result in the RANS equations becoming closed and hence allow them to be solved. The method by which the turbulent viscosity is calculated differs between turbulence models and will be the subject of the following sections.

2.5.2 Algebraic Turbulence Modelling

The first methods that successfully closed the RANS equations through the use of the Boussinesq hypothesis used algebraic expressions to evaluate the turbulent viscosity. These methods were in general designed for modelling two-dimensional turbulent boundary layer flows however they have been used more widely with some success. One such approach suggested by Prandtl[42], by analogy with kinetic theory, was the mixing length model. This states that μ_t is equal to the local product of the density,

the mean velocity gradient and the square of a characteristic length of the turbulent motions, l, the mixing length, i.e.

$$\mu_t = \rho l^2 \left| \frac{\partial U}{\partial y} \right| \qquad (2.51)$$

where, with reference to boundary layer theory, U is the mean velocity parallel to the freestream and y is normal to the wall. Clearly to complete the model the mixing length must be related to flow conditions which can be achieved through the following relationship

$$\begin{aligned} l &\approx \kappa y \left(1 - e^{-\frac{y^+}{A}} \right) && \kappa y \le 0.09\delta \\ l &\approx 0.09\delta && \kappa y > 0.09\delta \end{aligned} \qquad (2.52)$$

where $\kappa \approx 0.41$ is von Karman's constant, δ is the boundary layer thickness and A is a constant that varies with flow conditions, but for a smooth impermeable wall with zero pressure gradient is approximately 26. y^+ has units of length and is defined as yv^*/υ where v^* is the wall friction velocity and υ is the kinematic viscosity.

Spalding[43] attacked the problem of evaluating the eddy viscosity in the inner layer of a turbulent boundary layer without the mixing length concept. He arrived at the following relationship that also allowed him to derive his famous law of the wall.

$$\mu_{t,\text{inner}} \approx \mu\kappa e^{-\kappa B} \left(e^Z - 1 - Z - \frac{Z^2}{2} \right) \quad Z = \frac{\kappa U}{v^*} \qquad (2.53)$$

where $B \approx 5.0$ is a near-universal constant for flow past smooth impermeable walls. Clearly to complete Spalding's model a comparable expression is required for the eddy viscosity in the outer layer and one such expression was derived by Clauser[44] and is given below

$$\mu_{t,\text{outer}} \approx C\rho U_e \delta^* \qquad (2.54)$$

where $C \approx 0.016$, U_e is the freestream velocity and δ^* is the displacement thickness.

To find a solution to the RANS equations using any of the algebraic models detailed above requires some sort of iterative strategy. The main reason for this is that the relationships for μ_t in the inner layer are functions of ν^* which is unknown. However, another reason is the expression for the outer layer depends on δ for the mixing length model and δ^* for the Clauser model. To avoid this Baldwin and Lomax[45] suggested an alternative outer layer model that utilised a maximum function occurring in the boundary layer. This model has probably become the most popular of the algebraic turbulence models to date and takes the following form

$$\mu_{t,\text{outer}} \approx \frac{0.016 C_{cp} \rho y_{\max} F_{\max}}{1+5.5\left(C_{kleb}\, y / y_{\max}\right)^6} \qquad (2.55)$$

where

$$F_{\max} = \max\left[y\left|\frac{\partial U}{\partial y}\right|\left(1-e^{-\frac{y^+}{A}}\right)\right] \qquad (2.56)$$

and where y_{max} is the value of y corresponding to F_{max}. The parameters C_{cp} and C_{kleb} were originally taken to be constants of the order of unity however Granville[46] proposed that they should in fact be variable. He fitted them to known properties of Coles'[47] wake law and equilibrium pressure gradients and arrived at

$$C_{kleb} \approx \frac{2}{3} - \frac{0.01312}{0.1724+\beta^*} \qquad \beta^* = \frac{y_{\max}}{\nu^*}\frac{dU_e}{dx}$$
$$C_{cp} \approx \frac{3-4C_{kleb}}{2C_{kleb}\left(2-3C_{kleb}+C_{kleb}^2\right)} \qquad (2.57)$$

2.5.3 One Equation Turbulence Models

Prandtl[42] developed his mixing length theory by proposing that a more representative velocity scale in equation (2.51) would be the square root of the turbulent kinetic energy. This allows equation (2.51) to be recast in the following form

$$\mu_t = \rho k^{\frac{1}{2}} l \qquad (2.58)$$

Obviously, to evaluate equation (2.58) relationships for k and l must be found. Prandtl's previous work yields an expression for l, equation (2.52), but in order to calculate k some new analysis is required.

An exact transport equation for k can be found from the Reynold's stress equation. The derivation of the conservation relationship involves subtracting the RANS equation from it's instantaneous value, for both i and j directions. The ith result is multiplied by u_j' and added to the jth result multiplied by u_i'. This relation is then time averaged to yield the Reynolds stress equation, see Chen and Jaw[48],

$$\frac{\partial \rho U_m \overline{u_i' u_j'}}{\partial x_m} = \frac{\partial}{\partial x_m}\left(-\rho \overline{u_i' u_j' u_m'} - \overline{p'\left(\delta_{jm} u_i' + \partial_{im} u_j'\right)} + \mu \frac{\partial \overline{u_i' u_j'}}{\partial x_m} \right)$$

$$-\rho\left(\overline{u_i' u_m'} \frac{\partial U_j}{\partial x_m} + \overline{u_j' u_m'} \frac{\partial U_i}{\partial x_m} \right) - 2\mu \overline{\frac{\partial u_i'}{\partial x_m} \frac{\partial u_j'}{\partial x_m}} \qquad (2.59)$$

$$+ p'\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i} \right)$$

Setting $i=j$ in equation (2.59) yields the equation for turbulent kinetic energy i.e.

$$\frac{\partial \rho U_m k}{\partial x_m} = \frac{\partial}{\partial x_m}\left(-\frac{\rho}{2} \overline{u_i' u_i' u_m'} - \overline{p' u_m'} + \frac{\mu}{2} \frac{\partial \overline{u_i' u_i'}}{\partial x_m} \right) - \rho \overline{u_i' u_m'} \frac{\partial U_i}{\partial x_m} \qquad (2.60)$$

$$- \mu \overline{\frac{\partial u_i'}{\partial x_m} \frac{\partial u_i'}{\partial x_m}}$$

Equation (2.60) is extremely complex with many unknowns and therefore modelling ideas are required in order to find a solution to this equation and hence evaluate k.

The first term on the right hand side of equation (2.60) represents both the molecular and turbulent diffusion of k. It is reasonable to assume that the turbulent diffusion will dominate the molecular diffusion, see Libby[49], and is therefore sensible to only model the turbulent term. A widely used method of modelling the turbulent diffusion is to represent it in terms of gradients of k i.e.

$$\frac{\rho}{2}\overline{u'_m u'_i u'_i} + \overline{p' u_m} = -\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_m} \qquad (2.61)$$

where σ_k is an empirical constant. The production of k, represented by the second term on the right hand side of equation (2.60), can be modelled by invoking the Boussinesq hypothesis. The final term on the right hand side of equation (2.60) is the rate of turbulent dissipation of k defined by

$$\varepsilon = \frac{\mu}{\rho}\overline{\frac{\partial u'_i}{\partial x_m}\frac{\partial u'_i}{\partial x_m}} \qquad (2.62)$$

Applying these approximations and definitions to equation (2.60), an approximate transport equation for k can be written in the following form

$$\frac{\partial \rho U_i k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right) + G_k - \rho\varepsilon \qquad (2.63)$$

where G_k represents the generation of k and is defined as

$$G_k = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} \qquad (2.64)$$

Equation (2.63) can now be coupled with the Reynolds averaged continuity and Navier-Stokes equations and an expression for l to solve the closure problem. However, closure will not be obtained until the turbulent dissipation of k is modelled and this can be achieved by using the following relationship

$$\varepsilon = \frac{C_D k^{\frac{3}{2}}}{l} \qquad (2.65)$$

where C_D is an empirical constant. Predictions of turbulent flow using this one equation model, commonly known as Prandtl's k-l model are normally satisfactory, however they are often no better than the best of the algebraic turbulence models. Consequently one equation models are not at present popular because the added complexity of solving equation (2.63) does not yield significant benefits.

2.5.4 Two Equation Turbulence Models

Launder and Spalding[50] showed that the approximate turbulent kinetic energy equation arrives at a better approximation of μ_t if it is coupled with a comparable relationship for the dissipation, ε. If ε can be calculated then a combination of equations (2.58) and (2.63) will eliminate the length scale, l, and result in an appropriate expression for the turbulent viscosity i.e.

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \qquad (2.66)$$

where C_μ is an empirical constant.

An exact equation for the rate of dissipation of turbulent kinetic energy can be obtained by the following approach. Subtract the RANS equation from its instantaneous value and perform the operation $\partial/\partial x_m$ on the ith result. Multiply the resulting equation by $\partial u_i'/\partial x_m$ and then take the time average of this product and multiply by two. This results in the following exact transport equation for ε, see Chen and Jaw[48],

$$\frac{\partial \rho U_m \varepsilon}{\partial x_m} = \frac{\partial}{\partial x_m}\left(-\mu \overline{u_m \frac{\partial u_i'}{\partial x_m}\frac{\partial u_i'}{\partial x_m}} - \frac{2\mu}{\rho}\overline{\frac{\partial u_m'}{\partial x_j}\frac{\partial p'}{\partial x_j}} + \mu \frac{\partial \varepsilon}{\partial x_m} \right)$$

$$- 2\mu \overline{u_m' \frac{\partial u_i'}{\partial x_j}\frac{\partial^2 U_i}{\partial x_m \partial x_j}} - 2\mu \frac{\partial U_i}{\partial x_j}\left(\overline{\frac{\partial u_m'}{\partial x_i}\frac{\partial u_m'}{\partial x_j}} + \overline{\frac{\partial u_i'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}} \right) \qquad (2.67)$$

$$- 2\mu \overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}} - 2\frac{\mu^2}{\rho}\overline{\left(\frac{\partial^2 u_i'}{\partial x_m \partial x_m} \right)^2}$$

As with the RANS equation and the k transport equation, modelling is required to allow a solution to this equation to be found.

Again the first term on the right hand side of equation (2.67) represent the molecular and turbulent diffusion of ε. It is assumed that turbulent diffusion will

dominate molecular diffusion, see Libby[49], and with this assumption the term can be modelled in an analogous manner to equation (2.61) i.e.

$$\mu\overline{u_m \frac{\partial u_i'}{\partial x_m}\frac{\partial u_i'}{\partial x_m}} + \frac{2\mu}{\rho}\overline{\frac{\partial u_m'}{\partial x_j}\frac{\partial p'}{\partial x_j}} = -\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_m} \qquad (2.68)$$

where σ_ε is an empirical constant. The production of ε is represented by the second and third terms on the right hand side of equation (2.67). The magnitude of the first of these two terms can be assumed to be zero, see Chen and Jaw[48]. Whilst the second term was modelled by Hanjalic and Launder[51] by contracting indices to arrive at the following expression

$$2\mu\frac{\partial U_i}{\partial x_j}\left(\overline{\frac{\partial u_m'}{\partial x_i}\frac{\partial u_m'}{\partial x_j}} + \overline{\frac{\partial u_i'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}}\right) = C_1\rho\frac{\varepsilon}{k}\overline{u_i'u_j'}\frac{\partial U_i}{\partial x_j} = -C_1\frac{\varepsilon}{k}G_k \qquad (2.69)$$

where C_1 is an empirical constant. The final term on the right hand side of equation (2.67) represents the destruction of ε. This term was modelled by Hanjalic and Launder[51] as,

$$2\mu\overline{\frac{\partial u_i'}{\partial x_j}\frac{\partial u_i'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}} + 2\frac{\mu^2}{\rho}\overline{\left(\frac{\partial^2 u_i'}{\partial x_m \partial x_m}\right)^2} = C_2\frac{\varepsilon^2}{k} \qquad (2.70)$$

where C_2 is an empirical constant, on the basis that at a high turbulent Reynolds number, these two terms can taken as controlled by the dynamics of the energy cascade process transporting energy from lower to higher wave numbers. Substituting equations (2.68), (2.69) and (2.70) into equation (2.67) the approximate transport equation for ε can be expressed in the following form

$$\frac{\partial \rho U_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_1\frac{\varepsilon}{k}G_k - C_2\rho\frac{\varepsilon^2}{k} \qquad (2.71)$$

On the basis of experimental data from a wide variety of turbulent flows Launder et al. [52] recommended that the empirical coefficients C_1, C_2, C_μ, σ_k and σ_ε which appear in equations (2.63), (2.66) and (2.71) should take the values

$$C_1 = 1.44, C_2 = 1.92, C_\mu = 0.09, \sigma_k = 1.0 \text{ and } \sigma_\varepsilon = 1.3 \qquad (2.72)$$

Combining equations (2.63), (2.66) and (2.71) with the Reynolds averaged continuity and Navier-Stokes equations solves the closure problem and provides a complete system of equations for the analysis of turbulent flow. This system is called the k-ε model and has become the most widely used model for the calculation of turbulent flow by fluids engineers. However, the model is only applicable to high Reynold's number flow and because molecular viscosity and sub-layer damping effects are ignored, the model should only be used in the outer and overlap layers of a boundary layer flow.

In order to calculate turbulent flows near to a wall two options are available. Either so called wall functions can be used or an adapted form of the k-ε model, designed for low Reynold's number, can be implemented. The wall function approach essentially assumes that in near-wall regions the log-law holds. The k-ε calculations are then begun at a point y_p in the log-law region where the turbulent flow variables are calculated from the following relationships

$$\frac{U_p}{v^*} = \frac{1}{\kappa}\ln\left(\frac{v^* y_p}{\upsilon}\right) + B \quad k_p = \frac{v^{*2}}{C_\mu^{\frac{1}{2}}} \quad \varepsilon_p = \frac{v^{*3}}{\kappa y_p} \qquad (2.73)$$

A more detailed description of how wall functions are used in practice can be found in Chapter 3.

If more detailed information is required in the near wall region or a low Reynolds number flow is to be calculated then the low Reynolds number form of the k-ε model should be used. At lower Reynolds numbers the constants C_μ and C_2 become variable and in near wall regions the effects of molecular diffusion must be considered. Jones and Launder[53] and Launder and Sharma[54] suggested the following modifications to equations (2.63) and (2.71) to account for these factors

$$\frac{\partial \rho U_i k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu + \mu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right) + G_k - \rho\varepsilon + 2\rho\upsilon\left(\frac{\partial k^{\frac{1}{2}}}{\partial x_j}\right)^2 \qquad (2.74)$$

$$\frac{\partial \rho U_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu + \mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_1\frac{\varepsilon}{k}G_k - C_2\rho\frac{\varepsilon^2}{k} + 2\rho\upsilon\upsilon_t\left(\frac{\partial^2 U_j}{\partial x_m \partial x_i}\right) \qquad (2.75)$$

where the two extra terms with coefficient $2\rho\upsilon$ have been added for convenience to make ε vanish at the wall and whilst C_1, σ_k, and σ_ε take the same values as given in (2.72), C_μ and C_2 are redefined as

$$C_\mu = 0.09e^{-\frac{3.4}{(1+R_T/50)^2}} \qquad (2.76)$$

$$C_2 = 1.92\left(1 - 0.3e^{-R_T^2}\right) \qquad (2.77)$$

where the turbulent Reynolds number, R_T, is given by

$$R_T = \frac{k^2}{\varepsilon\upsilon} \qquad (2.78)$$

By applying these modifications Jones and Launder[53] showed that a variety of low Reynolds number flows could be predicted successfully. Subsequently other authors [55] have built on this foundation to offer improvements and additions however the underlying structure remains that initially proposed by [53].

The k-ε model in both it's high and low Reynolds number form has become the workhorse of practical engineering calculations. Despite numerous papers in the open literature offering improvements and modifications the fact still remains that for a large percentage of problems, the k-ε model will reliably produce results that, for the computational overheads associated with the model, are reasonably accurate. Clearly the model is not perfect and there are plenty of problems that violate it's modelling assumptions, resulting in inaccurate solutions. However, as yet when simplicity, accuracy and general applicability are consider the k-ε model has not been bettered.

2.5.5 Reynolds Stress Models

A much more complex approach to closing the RANS equations is to discard the Boussinesq's hypothesis and the concept of eddy viscosity and to solve the Reynolds stress equation, (2.59), directly for $\overline{u_i'u_j'}$. In order to present this approach consider equation (2.59) rewritten in the following form

$$\frac{\partial \rho U_m \overline{u_i'u_j'}}{\partial x_m} = D_{ij} + P_{ij} + \pi_{ij} - \varepsilon_{ij} + \mu \frac{\partial^2 \overline{u_i'u_j'}}{\partial x_m^2} \qquad (2.79)$$

where

$$\begin{aligned}
D_{ij} &= \text{diffusion} = \frac{\partial}{\partial x_m}\left(-\rho\overline{u_i'u_j'u_m'} - \overline{p'\left(\delta_{jm}u_i' + \partial_{im}u_j'\right)}\right) \\
P_{ij} &= \text{production} = -\rho\left(\overline{u_i'u_m'}\frac{\partial U_j}{\partial x_m} + \overline{u_j'u_m'}\frac{\partial U_i}{\partial x_m}\right) \\
\pi_{ij} &= \text{pressure - strain} = \underline{p\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)} \\
\varepsilon_{ij} &= \text{dissipation} = 2\mu\frac{\partial u_i'}{\partial x_m}\frac{\partial u_j'}{\partial x_m}
\end{aligned} \qquad (2.80)$$

Clearly, in order to solve equation (2.79) each term must be expressed in terms of variables for which we have equations. If equation (2.79) is coupled with continuity, the RANS equations and transport equations for k and ε then the available variables are U_i, p, k, ε and $\overline{u_i'u_j'}$. The production term is already expressed in these terms, but the other three terms require modelling. Following the approach suggested by Rodi[56] the diffusion, pressure-strain and dissapation terms can be modelled with the following expressions

$$D_{ij} = C_s \frac{\partial}{\partial x_n}\left(\frac{k}{\varepsilon}\overline{u_m'u_n'}\frac{\partial \overline{u_i'u_j'}}{\partial x_m}\right) \qquad (2.81)$$

$$\pi_{ij} = -C_1\frac{\varepsilon}{k}\left(\overline{u_i'u_j'} - \frac{2}{3}k\delta_{ij}\right) - C_\gamma\left(P_{ij} - \frac{1}{3}P_{ii}\delta_{ij}\right) \qquad (2.82)$$

$$\varepsilon_{ij} = \frac{2\varepsilon}{3}\delta_{ij} \qquad (2.83)$$

where the suggested empirical constants are

$$C_S = 0.25, C_1 = 1.5 \text{ and } C_\gamma = 0.6 \qquad (2.84)$$

Despite the Reynolds stress approach being recognised as an improvement over coupling eddy viscosity with transport equations for k and ε it has not found wide application. The chief reason for this is the large system of equations that must be solved. For example in three dimensions a total of twelve partial differential equations must be solved which inevitably becomes computationally complex and time consuming. Therefore, beginning with Hanjalic and Launder[57] considerable effort has been devoted to developing algebraic Reynolds stress models. For example, Rodi[56] suggests the following formula

$$\overline{u_i'u_j'} = \frac{2}{3}k\delta_{ij} + \frac{\left(1-C_\gamma\right)\left(\frac{k}{\varepsilon}\right)\left(P_{ij} - \frac{1}{3}P_{ii}\delta_{ij}\right)}{C_1 + \left(\frac{1}{2\varepsilon}\right)P_{ii} - 1} \qquad (2.85)$$

The computational advantage of equation (2.85) over (2.79) is obvious. Nevertheless, as computational power increases and programming languages improve the barriers to using equation (2.79) are being removed and turbulent calculations will undoubtedly make more use of the Reynolds stress model in the future.

2.5.6 Large Eddy Simulation

Although a large percentage of the research carried out into turbulence modelling in the last thirty years has been based around the idea of time averaging it has not been the only approach. Another approach that has become more popular as computing power has increased is large eddy simulation (LES). In this method, rather than averaging turbulent quantities over time instead they are averaged in space. Essentially a filtering method is used such that small turbulent eddies, i.e. those that are too small to resolve over a typical grid scale, are removed from the flow field in

such a way that the governing equations only come to apply to the larger resolvable eddies. The bases for this approach is that large eddies interact most strongly with the mean flow and therefore are mainly responsible for the transport of momentum and energy by advection. Also, geometry and flow type dictate the large eddy structure, which is often highly anisotropic and their time scales are similar to those of the mean flow.

In contrast, the smaller irresolvable eddies are mainly produced by non-linear interactions with the large eddies and therefore only influence the large eddies themselves and not the mean flow directly. Their time scales are much smaller than the mean flow and they are less dependent on geometry and much closer to isotropy. Therefore, while the large eddy structure normally varies between flows it is possible that a general model can be created for the small eddy structure or at least that any model will have a lesser influence on the calculation as a whole.

The small scale eddies are removed from the flow field by applying the following convolution operator to the governing continuity and Navier-Stokes equations

$$\overline{f}(\mathbf{x}) = \iiint_{-\infty}^{\infty} G(\mathbf{x}-\mathbf{x}')f(\mathbf{x}')d\mathbf{x}' \qquad (2.86)$$

where the overbar refers to a spatial volume-averaged value of $f(\mathbf{x})$ and $G(\mathbf{x})$ is a filtering function that satisfies the following requirements

- $\iiint_{-\infty}^{\infty} G(\mathbf{x})d\mathbf{x} = 1$
- All moments of $G(\mathbf{x})$ exist.
- The convolution operator (2.86) produces a "smoothed" version of $f(\mathbf{x})$.
- The width of the filter function $G(\mathbf{x})$ over which most of the integration takes place is comparable to the shortest resolvable wavelength of the calculation grid i.e. for a grid of size $\Delta\mathbf{x}$ this would be $2\Delta\mathbf{x}$

The most widely used filter function in LES is the Gaussian filter given by

$$G(\mathbf{x}-\mathbf{x}')=\sqrt{\frac{6}{\pi}}\left[\frac{1}{2\mathbf{x}}e^{-\frac{6(\mathbf{x}-\mathbf{x}')^2}{(2\Delta\mathbf{x})^2}}\right] \qquad (2.87)$$

By applying the convolution operator to the continuity and Navier-Stokes equations, see [58] for details, it can be shown that the filtered form of the governing equations is

$$\frac{\partial \overline{u_i}}{\partial x_i}=0 \qquad (2.88)$$

$$\frac{\partial \rho\overline{u_i}}{\partial t}+\frac{\partial \rho\overline{u_i u_j}}{\partial x_j}=-\frac{\partial \overline{p}}{\partial x_i}+\mu\frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} \qquad (2.89)$$

where the overbar indicates a space filtered or averaged variable. As in time averaging, equations (2.88) and (2.89) present a closure problem and again it is the non-linear term $\overline{u_i u_j}$ that causes the difficulty. In order to resolve this problem consider, in an analgous manner to time averaging, the velocity field separated into resolvable and irresolvable components i.e.

$$u_i=\overline{u_i}+u_i' \qquad (2.90)$$

where $\overline{u_i}$ is the filtered resolvable velocity field and u_i' is the irresolvable or subgrid-scale velocity field. Substituting equation (2.90) into the non-linear term and expanding we arrive at

$$\overline{u_i u_j}=\overline{\overline{u_i}\,\overline{u_j}}+\overline{\overline{u_i}u_j'}+\overline{u_i'\overline{u_j}}+\overline{u_i'u_j'} \qquad (2.91)$$

Clearly the first term on the RHS of equation (2.91) can be evaluated from the resolvable velocity components whereas the remaining terms all contain irresolvable components and therefore we must again turn to modelling ideas.

Consider then the subgrid scale stresses (SGS) defined as

$$R_{ij}=\overline{\overline{u_i}u_j'}+\overline{u_i'\overline{u_j}}+\overline{u_i'u_j'} \qquad (2.92)$$

or

$$R_{ij} = \overline{u_i u_j} - \overline{\overline{u_i u_j}} \tag{2.93}$$

Furthermore define the off-diagonal subgrid scale shear stresses as

$$\tau_{ij} = R_{ij} - \frac{1}{3}\delta_{ij} R_{mm} \tag{2.94}$$

and also a modified pressure as

$$\frac{P}{\rho} = \frac{\overline{p}}{\rho} + \frac{1}{3} R_{mm} \tag{2.95}$$

Substituting equations (2.94) and (2.95) into equation (2.89) gives

$$\frac{\partial \rho \overline{u_i}}{\partial t} + \frac{\partial \rho \overline{\overline{u_i u_j}}}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \mu \frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.96}$$

The filtered Navier-Stokes equations are now in a form that more readily lends itself to modelling of the SGS however before moving onto to modelling these stresses consider the advection term on the RHS of equation (2.96). As Leonard[59] first discussed this term contains the filtered product $\overline{\overline{u_i u_j}}$, but all that can really be resolved on a given grid is $\overline{\overline{u_i}\,\overline{u_j}}$. Therefore following [58], let the Leonard stress λ_{ij}, be defined as

$$\lambda_{ij} = \overline{\overline{u_i u_j}} - \overline{\overline{u_i}\,\overline{u_j}} \neq 0 \tag{2.97}$$

An approximate method to evaluate the Leonard stress using a Taylor series expansion and an isotropic Gaussian filter with effective filter width Δx_m gives, see [58]

$$\lambda_{ij} = \frac{(\Delta x_m)^2}{6} \frac{\partial^2 \left(\overline{u_i u_j} \right)}{\partial x_m \partial x_m} + \text{HOT} \tag{2.98}$$

Leslie and Quarini[60] show that for a typical high Reynolds number flow the Leonard stress accounts for approximately 14% of the total energy transfer from large eddies to subgrid scales. Consequently, the advection term in equation (2.96) should be taken as

$$\frac{\partial}{\partial x_j}\left(\overline{u_i u_j} + \lambda_{ij}\right) \qquad (2.99)$$

Returning now to the SGS, a model is required that represents the transfer of energy from the large to small eddies. Generally, this transfer is in the form of a loss or dissipation of energy from the large eddies and therefore any SGS model should be dissipative in nature. However, although the net energy transfer is to the small eddies, in fact energy is transferred in both directions and consequently whilst the SGS model should be dissipative it should also model the effect of two way energy transfer on the large scale motion. As in time averaging most SGS modellers have relied on a simple eddy viscosity approach to represent R_{ij}. This model was first introduced by Smagorinsky[61] and assumes the SGS are proportional to the modulus of the strain rate tensor of the filtered large scale flow i.e.

$$\tau_{ij} = 2\upsilon_t \overline{S_{ij}} = \upsilon_t \left| \frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right| \qquad (2.100)$$

where the turbulent kinematic viscosity is given by the following relationship

$$\upsilon_t = a_1 h^2 \overline{S}^{\frac{1}{2}} \qquad (2.101)$$

and

$$\overline{S} = \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) \frac{\partial \overline{u_i}}{\partial x_j} \qquad (2.102)$$

$$h = 8^{\frac{1}{3}} \left(\Delta x_1 \Delta x_2 \Delta x_3 \right)^{\frac{1}{3}} \qquad (2.103)$$

$$a_1 = 0.176 \qquad (2.104)$$

So in summary the complete set of equations for the resolvable velocity components are

$$\frac{\partial \overline{u_i}}{\partial x_j} = 0 \qquad (2.105)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho\left(\overline{u_i u_j} + \lambda_{ij}\right)\right] = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\left(\mu + \mu_t\right)\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right)\right] \qquad (2.106)$$

with μ_t and λ_{ij}, for a Gaussian filter, evaluated using equations (2.98) and (2.106) respectively.

Whilst LES techniques undoubtedly provide a greater insight into the internal workings of turbulent flow they are not necessarily the ideal choice of turbulence model for practical engineering applications. In most instances what is required for engineering applications is precisely what time averaging provides i.e. time mean values for the primary flow variables. Information on the intricacies of turbulent eddies, that LES with it's space averaging methods provides, is not required as the interpretation of this information as an input into a design process is too complex. However, LES techniques can be utilised to develop better, more practical time average techniques that will be of greater use in realistic engineering applications.

2.6 Grid Generation

CFD techniques, whether they utilise finite difference, finite volume or finite element methods, are based on the spatial subdivision of a domain into polyhedra immediately implying the need for generating a mesh. Mesh types are as varied as the numerical methods that they accompany and can be split into a number of different groups. The coarsest classification is whether a mesh is regarded as structured or unstructured. As the name indicates structured meshes adhere to a regular pattern which will typically be constructed from quadrilaterals in two dimensions or hexahedra in three dimensions. A more precise definition of a structured mesh is that each point within the grid has the same number of neighbours and consequently it's topology can be stored in a logical matrix structure. In contrast in an unstructured mesh, generally constructed from triangles and tetrahedra in two and three dimensions respectively, each point will have an arbitrary number of neighbours.

After the coarse structured/unstructured categorisation, mesh types can be further classified by their conformity, surface or body alignment and element type. Conformity denotes the continuity of neighbouring elements across edges in two dimensions and faces in three dimensions. In other words a conformal mesh element will have the same number of neighbouring elements as it does faces whilst in a non-conformal mesh this will not necessarily be the case. Surface or body alignment is achieved in a mesh when boundary faces or edges perfectly match the surface of the domain to be meshed. Non-aligned grids in contrast have surfaces that cross the domain boundary surface. Finally, element type describes the polyhedra that are used to discretise the domain of interest. Typical types of element are quadrilaterals and triangles in two dimensions and hexahedra and tetrahedra in three dimensions, as has already been mentioned.

Here, as it is the form of mesh that the discretisation strategy presented in this thesis is designed for, the focus will be on grid generation of unstructured, conformal, surface aligned grids consisting of triangles and tetrahedra. There are two main methods of generating this form of grid, the advancing front technique[62] and Delaunay triangulation[63]. Both methods are popular and neither has come to the fore as being superior to other as yet. Nevertheless, in this thesis all unstructured grid generation has been performed using Delaunay triangulation because of the availability of the commercial grid generator Gambit which utilises the Delaunay triangulation technique.

2.6.1 The Advancing Front Technique

The aim of the advancing front technique is to proceed from the boundary of the computational domain adding elements to the unmeshed space until the domain has been completely filled with elements. The 'front' is the boundary between the region of space that has been meshed and that which is still empty. Algorithmically the advancing front technique consists of the following steps

1. Define the boundaries of the domain to be meshed.

2. Define the spatial variation of element size and shape i.e. areas where the mesh should be coarser or finer and the aspect ratio elements should have.

3. Using the information given for the distribution of element size and shape generate sides along the surface patches that make up the boundary surface. These form the initial front for the triangulation of the boundary surface.

4. Using the information given for the distribution of element size and shape, and the boundary surface definition triangulate the boundary surfaces. This yields the initial front of faces.

5. Find the generation parameters (element size and shape) for these faces

6. Select the next face to be deleted from the front. To avoid large elements crossing over regions of small elements, the face forming the smallest new element is selected as the next face to be deleted from the list of faces.

7. For the face selected to be deleted

    a. Select a 'best point' position for the introduction of a new point.

    b. Determine whether a point exists in the already generated grid that should be used in lieu of the new point. If there is such a point make this the new point and continue searching (go to 7b).

    c. Determine whether the element formed with the selected new point crosses any given faces. If it does, select another new point and try again (go to 7c).

8. Add the new element, point and faces to their respective lists.

9. Find the new generation parameters for the new faces.

10. Delete the known faces from the list of front faces.

11. If there are any faces left in the front, go to 6.

What appears on paper to be a relatively straight forward algorithm is in fact complex to code efficiently for a computer. One of the key issues is the necessity to use efficient data structures that allow the grid connectivity data to be searched quickly. Also, a reliable and fast method for checking whether two faces intersect is

essential if the algorithm is to work in a reasonable time frame. More details on these issues and others which are beyond the scope of this thesis can be found in [64].

2.6.2 Delaunay Triangulation

Given a set of points $P := \mathbf{x}_1, \mathbf{x}_2, \ldots\ldots, \mathbf{x}_n$, one may define a set of regions or volumes $V := v_1, v_2, \ldots\ldots, v_n$ assigned to each of the points that satisfy the following property. Any location within v_i is closer to $\mathbf{x}_i$ than to any other of the points i.e. within v_i

$$\left|\mathbf{x} - \mathbf{x}_i\right| < \left|\mathbf{x} - \mathbf{x}_j\right| \forall\ j \neq i \qquad (2.107)$$

This set of volumes V, which covers the complete domain, is known as the Dirchlet Tessellation. The volumes v_i are convex polyhedra and are known as Voronoi regions. If all the pairs of points $\mathbf{x}_i$ and $\mathbf{x}_j$ across the polyhedra boundaries are joined then this forms the set of triangles commonly called the Delaunay triangulation. The set of triangles, or tetrahedra in three dimensions, satisfy the condition that no other point is contained within the circumcircle (circumsphere in 3D) formed by the nodes of the triangle. A number of practical triangulation procedures based upon the Delaunay method are evident in the open literature however most are based upon Bowyer-Watson[65,66] algorithm summarised below for three-dimensional domains.

1. Define the convex hull within in which all points will lie. This can either be a large single tetrahedron or a hexahedron subdivided into five tetrahedral.

2. Introduce a new point $\mathbf{x}_{n+1}$

3. Find all tetrahedra whose circumsphere contains $\mathbf{x}_{n+1}$. These tetrahedra will be deleted.

4. Find all points belonging to these tetrahedra.

5. Find all the external faces of the void that result from the deletion of these tetrahedra.

6. Form new tetrahedra by joining all the external faces found in 5 to the new point $\mathbf{x}_{n+1}$

7. Add the new elements and points to their respective lists

8. Update all data structures

9. If more points are to be introduced go to 2

The algorithm described above assumes a given point distribution, which can be a tedious task and may be replaced by a more general specification of desired element size and shape in space. A method for achieving this was proposed independently by Frey[67] and Holmes[68] which is detailed below.

1. Assume that a boundary point distribution is given

2. Generate a Delaunay triangulation of the boundary points

3. Using the information provided on element size and shape compute the desired element size and shape for the points of the current mesh.

4. Loop over the elements

   a. Add a new point at the centre of each element

   b. Compute the distances from the new point to the four nodes of the element.

   c. Compare the computed distances with the desired size and shape

   d. If any of the distances are smaller than α, a fraction of the desired element length skip the element and return to 4

   e. Compute distances from the new point to the other new points in the neighbourhood

   f. If any of the distances are smaller than β, a fraction of the desired element length skip the element and return to 4

   g. Store the desired element size and shape for the new point

5. Perform a Delaunay triangulation for the new points.

As with the advancing front technique coding this algorithm for a computer is not straightforward. The reasons for this are also very similar to those for the advancing front technique and again more details can be found in the open literature[64].

2.7 Conclusions

An overview of the relevant literature to the development of a cell centred, finite volume based algorithm for both laminar and turbulent flow, which facilitates the use of high order convective models has been presented in this chapter. The key areas covered were solution and discretisation methods, convective modelling, turbulence modelling and grid generation. This has given an insight into the problems faced in all these areas of research and identified how they are relevant to the solution algorithm developed here. As highlighted in Chapter 1 this thesis now intends to build upon this existing work and develop it concentrating particularly in the area of high order convective modelling. The following chapters will detail this and present the results and conclusions of this research.

CHAPTER 3

3 Discretisation and Solution Procedure

3.1 Introduction

The survey of the open literature presented in Chapter 2 indicates that whilst cell centred finite volume methods for the solution of the incompressible Navier-Stokes equations are well established on structured meshes their extension to unstructured grids is a relatively recent innovation[4, 5, 6]. In this chapter a cell centred finite volume discretisation of the governing equations is presented that is based on the work of Davidson[4] and Mathur and Murthy[5]. Both [4] and [5] used second order discretisations of the convective and diffusive terms in the governing equations. Here, in an analogous manner to Leonard's[17] QUICK scheme a second order discretisation of the diffusive term is coupled with an upwind third order convective model that has been adapted from its original use as part of a time-marching algorithm.

The solution strategy used is the SIMPLE algorithm, which, after its development by Patankar[15] twenty years ago, has become one of the standard method for solving the incompressible Navier-Stokes equations. To avoid the complication of using an unstructured staggered grid the pressure and velocity fields are prevented from de-coupling by using an interpolation scheme based on the method developed by Rhie and Chow[24]. In order to model turbulent flows the standard high Reynolds number k-ε turbulence model[50] is used along with wall functions[52] to avoid excessive grid refinement near wall boundaries. A brief overview of the governing equations for both laminar and turbulent flow is given at the beginning of this chapter. This is followed by the discretisation of the general governing differential equation and a description of the solution algorithm. The chapter concludes with a discussion of the most commonly found boundary conditions and how they are applied to the discretised equations.

3.2 Governing Equations

The conservation laws of mass and momentum govern steady, incompressible viscous flow. For a Newtonian fluid without body forces in a Cartesian co-ordinate system these laws are described by the following equations

$$\frac{\partial}{\partial x_i}(\rho u_i) = 0 \qquad (3.1)$$

$$\frac{\partial}{\partial x_j}\left(\rho u_i u_j - \mu \frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} \qquad (3.2)$$

where u_i, p, ρ and μ are velocity, pressure, density and viscosity respectively. Whilst the Reynolds number of a given flow is low, the flow will remain laminar and equations (3.1) and (3.2) can be solved directly. Conversely at high Reynolds number the smooth, orderly laminar flow will give way to the fluctuating disorderly motion of turbulence. All investigations[39] indicate that equations (3.1) and (3.2) do apply to turbulent flow. However, the wide range of flow scales involved would require solutions to be sought on meshes that are prohibitively large for the standard computing resources available today.

A number of solutions to the problem of calculating turbulent flows have been proposed and these have been reviewed in Chapter 2. Here, the time averaging philosophy coupled with the two equation k-ε turbulence model will be used to resolve the problem. This method requires that the continuity and momentum equations are recast in the following time-averaged form

$$\frac{\partial}{\partial x_i}(\rho U_i) = 0 \qquad (3.3)$$

$$\frac{\partial}{\partial x_j}\left[\rho U_i U_j - (\mu + \mu_t)\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] = -\frac{\partial P}{\partial x_i} \qquad (3.4)$$

where U_i and P are the time averaged components of velocity and pressure respectively and μ_t is the turbulent viscosity given by

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \qquad (3.5)$$

where $C_\mu = 0.09$, k and ε are an empirical constant, turbulent kinetic energy and it's rate of dissipation respectively. Clearly, expressions for k and ε are also required to close this set of equations and to that end the following transport relationships are used

$$\frac{\partial \rho U_i k}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right) + G_k - \rho\varepsilon \qquad (3.6)$$

$$\frac{\partial \rho U_i \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_i}\right) + C_1\frac{\varepsilon}{k}G_k - C_2\rho\frac{\varepsilon^2}{k} \qquad (3.7)$$

where $C_1 = 1.44$, $C_2 = 1.92$, $\sigma_k = 1.0$ and $\sigma_\varepsilon = 1.3$ are empirical constants and G_k is given by

$$G_k = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} \qquad (3.8)$$

In conclusion then in order to solve laminar flow problems equations (3.1) and (3.2) suffice whilst for turbulent problems a solution of equations (3.3) to (3.7) is required.

3.3 The Pointer System

The control volumes of an unstructured grid can be placed in a domain in a very irregular fashion, in contrast to a structured grid, which reflects some type of consistent geometrical regularity, see Figure 3.1. Without this regularity it is necessary to have a pointer system that explicitly stores information on grid topology, rather than using the implicit line structure that a structured mesh provides. The system used here is based on the fact that the cells, faces and vertices in the grid are all numbered separately and that the Cartesian co-ordinate of each vertex is known. Five lists combine to make up the pointer system that contains all the

necessary connectivity and boundary condition data required to apply the finite volume method to equations (3.1) to (3.7).

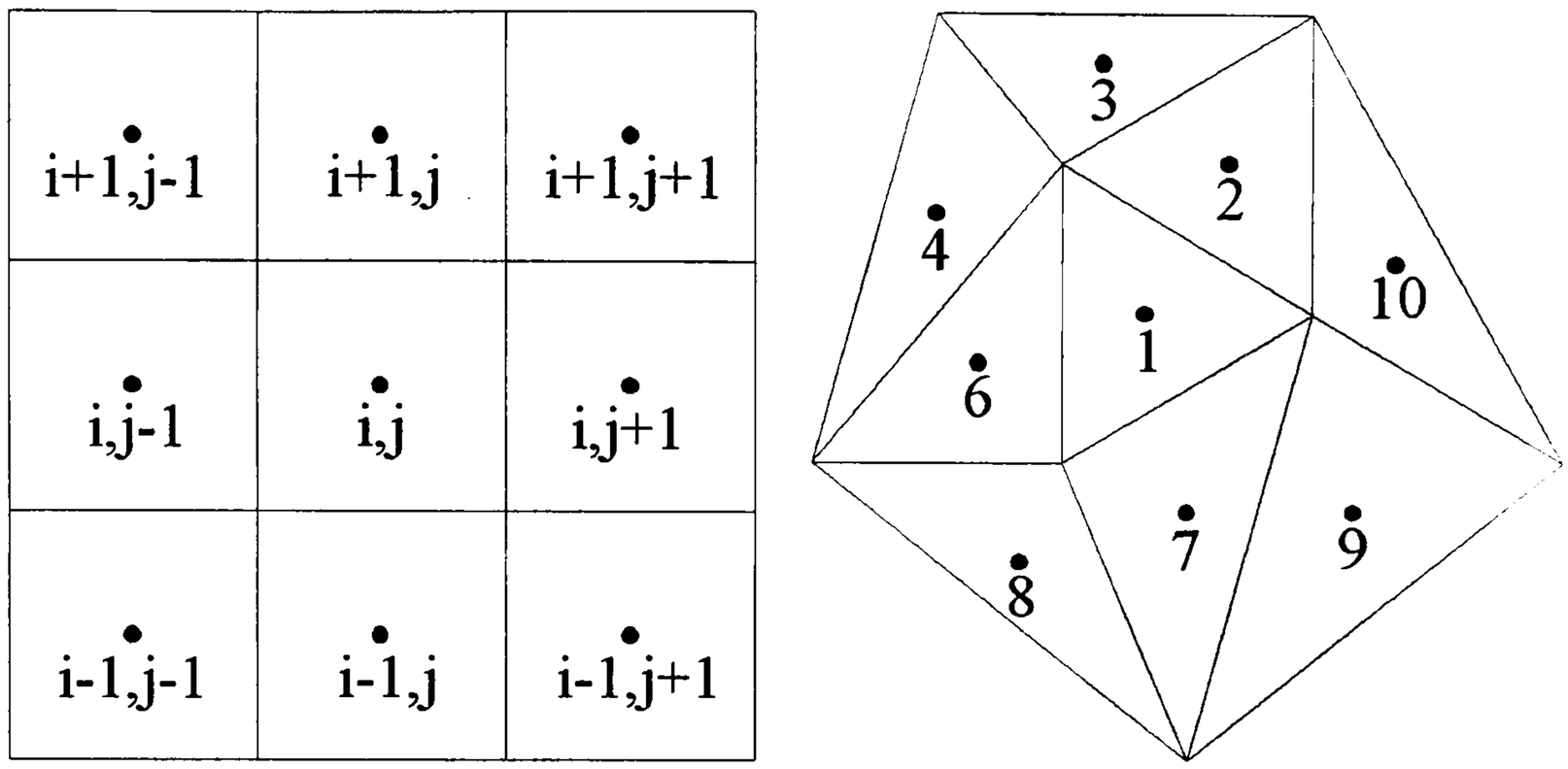


Figure 3.1: Comparison of the Grid Data Structure between Structured and Unstructured Meshes

3.3.1 LISTC

When solving discretised equations a pointer is required from a cell to its neighbouring cells. A cell's neighbours are those cells that are adjacent to it. This information is stored in LISTC, which has the following form

LISTC(i, 1)=total number of neighbouring cells

LISTC(i, 2)=index of neighbour 1

LISTC(i, 3)=index of neighbour 2

.

.

LISTC(i, LISTC(i,1)+1)=index of neighbour LISTC(i, 1)

where i is the index of the cell. It is necessary to store the total number of neighbours because grids can be made up of a variety of types of cells and therefore the number of neighbours will not be constant.

3.3.2 LISTCF

In analogous manner to LISTC, LISTCF points to the indices of the faces that go to make up a cell. Again this information is a requirement when solving discretised equations on an unstructured grid. LISTCF has the following form

LISTCF(i, 1)=index of face 1

LISTCF(i, 2)=index of face 2

.

.

LISTCF(i, LISTC(i,1))=index of face LISTC(i, 1)

where i is the index of the cell. Clearly there is no need to store the total number of faces that a particular cell comprises of, as it will be identical to the total number of neighbours.

3.3.3 LISTF

It is necessary to identify the different types of boundary that surround a domain so that appropriate conditions can be applied to the faces that make up those boundaries. Also, the discretisation process requires that a number of geometric quantities be known about a particular face. Therefore it is important that a faces type, the vertices that are joined to construct a face and the geometric shape of a face are all known. This information is stored in LISTF, which has the form

LISTF(i, 1)=index of vertex 1

LISTF(i, 2)=index of vertex 2

LISTF(i, 3)=index of vertex 3

LISTF(i, 4)=index of vertex 4

LISTF(i, 5)=type

LISTF(i, 5)=geometric shape

where i is the index of the face. Geometric shape can take three values each representing a particular kind of face. These are defined as

2=line

3=triangle

4=quadrilateral

Line is the only face type used in two-dimensional grids whereas triangle and quadrilateral can both be used in three-dimensional meshes. Similarly, type takes seven different values that are defined as

1=internal

2=wall

3=inlet

4=outlet

6=symmetry

8=periodic

12=periodic-shadow

Clearly internal represents all those faces that are not on the boundaries whilst the other types are specific forms of boundary, more details of which will be given later in this chapter. The number of indices specified in the first four storage areas of LISTF is obviously dependent on the geometric shape of the face and therefore a simple rule that a storage area not in use should be set to zero is useful to avoid confusion. Although LISTF is limited to only three shapes it could be easily adapted to include more. However, the three defined shapes are the most commonly used in unstructured mesh generation and are adequate for the purposes of this research.

3.3.4 LISTS

Some forms of finite volume discretisation require that not only is there a pointer from a cell to its neighbours but also one that points to the cells that surround it. Cell *a* would surround cell *b* if cell *a* was not a neighbour to cell *b* but a neighbour of cell *a* was one of the neighbours to cell *b*. For example in Figure 3.1 cells 3,4,8,9 and 10 surround cell 1. LISTS stores this information in a manner identical to LISTC i.e.

LISTS(*i*, 1)=total number of surrounding cells

LISTS(*i*, 2)=index of surrounding cell 1

LISTS(*i*, 3)=index of surrounding cell 2

.

.

LISTS(*i*, LISTS(*i*,1)+1)=index of surrounding cell LISTS(*i*, 1)

3.3.5 LISTE

The calculation of the volume of a cell is integral to forming discretised equations. Consequently the geometric shape of the cell itself is necessary information in order that an appropriate formula for the calculation of its volume can be used. This information is stored in LISTE, which has the form

LISTE(*i*)=geometric shape

where *i* is the index of the cell. Here geometric shape can take four values defined as,

1=triangle

2=quadrilateral

3=tetrahedron

4=hexahedron

As with the definition of geometric shape in LISTF some of the shapes defined for LISTE are only applicable to two-dimensional grids whilst others only apply to three-dimensional meshes.

3.3.6 LISTP

If periodic boundary conditions are included in a domain then LISTP is also required to store additional information about this particular type of boundary condition. Leaving a more detailed explanation of periodic boundaries to later in this chapter it will suffice to say that there are linked pairs of cells either side of a periodic boundary and LISTP stores this information in the following form

LISTP(i, 1)=index of first cell in pair

LISTP(i, 2)=index of second cell in pair

where i is the index of the periodic pair, which is an addition to the numbering of the cells, face and vertices of the grid.

3.4 Discretisation of the General Convection-Diffusion Equation

Equations (3.2), (3.4), (3.6) and (3.7) indicate that the dependent variables u_i, U_i, k and ε all obey a generalised conservation principle. If the dependent variable is denoted by ϕ then the general differential equation is given by

$$\frac{\partial}{\partial x_i}\left(\rho U_i\phi - \Gamma\frac{\partial\phi}{\partial x_i}\right) = S \qquad (3.9)$$

The three terms in equation (3.9) from left to right represent the convection term, the diffusion term and the source term. Depending on which variable ϕ stands for the diffusion coefficient Γ will take the values μ, $\mu+\mu_t$, μ_t/σ_k or μ_t/σ_ε whilst the source term S will be $-\partial p/\partial x_i$, $-\partial P/\partial x_i$, $G_k-\rho\varepsilon$ or $C_1G_k\varepsilon/k-C_2\rho\varepsilon^2/k$. The convection term in equation (3.9) has been written in terms of the time averaged velocity U_i. This is correct for the time averaged momentum equation and the k and ε transport equations, however for the laminar flow momentum equation U_i would be replaced with the instantaneous velocity u_i. The mean velocity will be used without loss of

generality in the following derivation. Therefore, in vector notation equation (3.9) becomes

$$\nabla.(\rho \mathbf{U}\phi - \Gamma\nabla\phi) = S \qquad (3.10)$$

Defining a total flux vector **J** as

$$\mathbf{J} = \rho \mathbf{U}\phi - \Gamma\nabla\phi \qquad (3.11)$$

equation (3.10) can be written in the form

$$\nabla.\mathbf{J} = S \qquad (3.12)$$

Following the finite volume method equation (3.12) is then integrated over a volume V to give

$$\int_V \nabla.\mathbf{J}dV = \int_V SdV \qquad (3.13)$$

Consider now the divergence theorem, it states that the outward flux of a vector field **F** across an orientated surface A with outward unit normal **n** equals the integral of the divergence of **F** over the volume V enclosed by the surface i.e.

$$\int_A \mathbf{F}.\mathbf{n}dA = \int_V \nabla.\mathbf{F}dV \qquad (3.14)$$

Applying the divergence theorem to the left hand side of the equation (3.12) we arrive at the following integral relation

$$\int_A \mathbf{J}.\mathbf{n}dA = \int_V SdV \qquad (3.15)$$

In this form the general differential equation can now be discretised by replacing the integrals with approximate algebraic expressions.

Consider the control volumes shown in Figure 3.2. Assuming that control volume C0 is taken as V in equation (3.15) then the left hand side of this equation can be approximated by

$$\int_A \mathbf{J}.\mathbf{n}dA = \sum_f \mathbf{J}_f.\mathbf{A}_f \qquad (3.16)$$

where $\mathbf{J}_f$ is a flux vector that is assumed to prevail over the whole of face f, $\mathbf{A}_f$ is the area of f multiplied by its normal outward pointing vector and the summation is over all the faces of the control volume. An approximation to the right hand side of equation (3.15) can be made in an analogous manner. Assuming that a constant value of the source term prevails throughout the control volume then the right hand side of equation (3.15) can be replaced by

$$\int_V S dV = S_0 \Delta V_0 \qquad (3.17)$$

where S_0 is the constant value of the source term in C0 and ΔV_0 is the volume of C0.

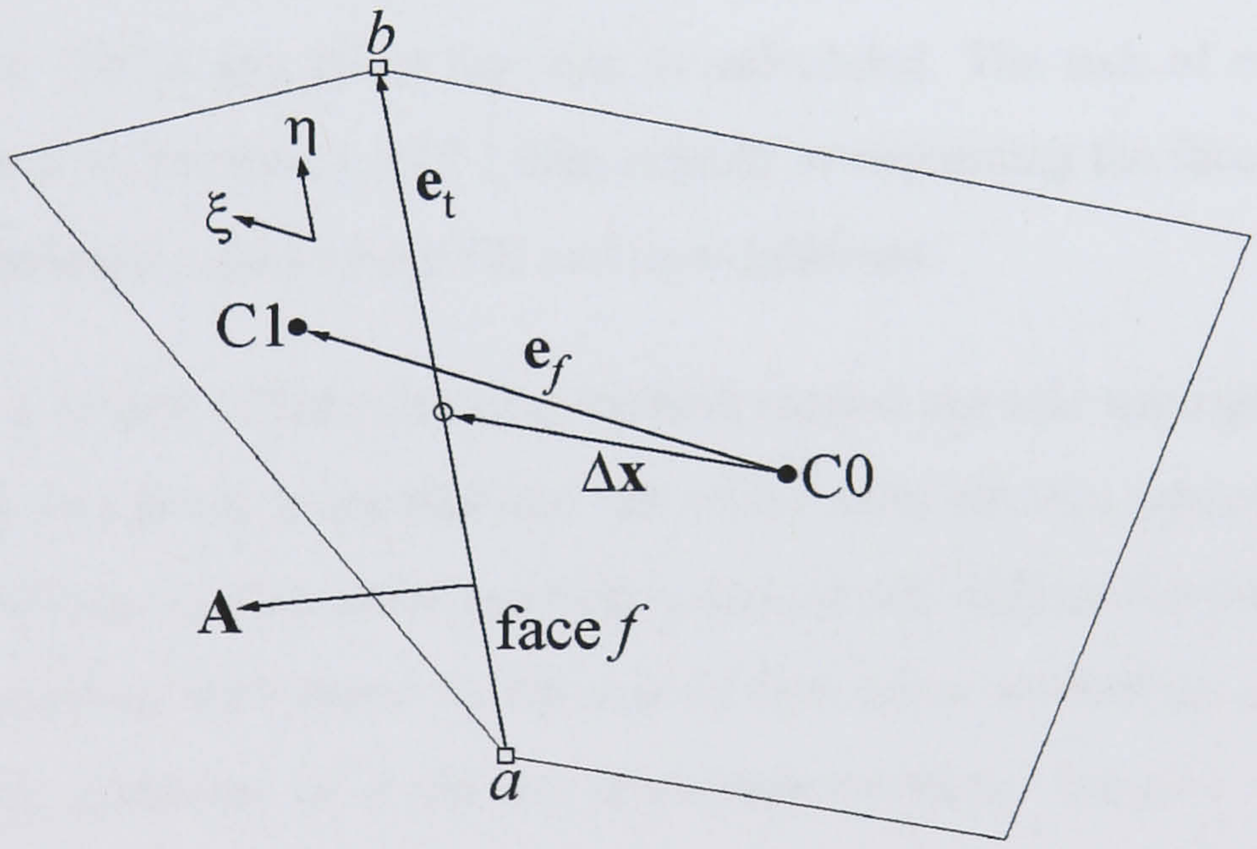


Figure 3.2: Control Volume Schematic. Solid circles denote nodes, open circles denote face centres and open squares denote vertices.

To complete the discretisation of equation (3.15) the approximate expressions (3.16) and (3.17) must be expressed in terms of values of ϕ at the nodes of the neighbouring control volumes. To facilitate this discussion the approximate expression (3.16) is expanded as follows

$$\sum_f \mathbf{J}_f.\mathbf{A}_f = \sum_f \left(\rho \mathbf{U}\phi - \Gamma \nabla \phi\right)_f.\mathbf{A}_f = \sum_f m_f \phi_f - D_f \qquad (3.18)$$

where $m_f=(\rho \mathbf{U}.\mathbf{A})_f$ is the mass flow rate across f and $D_f=(\Gamma \nabla \phi.\mathbf{A})_f$ is the transport of ϕ through f due to diffusion. Substituting equations (3.16), (3.17) and (3.18) into equation (3.15) gives the following relation

$$\sum_f m_f \phi_f - D_f = S_0 \Delta V_0 \qquad (3.19)$$

In this form the three distinct terms of the general differential equation can be seen. Again from left to right we have the convection term, the diffusion term and the source term. How each of these terms is expressed as a function of the node values of ϕ in C0 and the neighbouring control volumes will now be dealt with separately.

3.4.1 Convection Term

Assume for the moment that the velocity and density fields are known and therefore the mass flow rate at any given face can be calculated. The task of evaluating the convective term in equation (3.19) then reduces to expressing the face value ϕ_f as a function of the node values of ϕ at C0 and its neighbours.

In Chapter 2 a review of the extensive research carried out into appropriate schemes to evaluate ϕ_f was given. It showed that two of the most obvious schemes that could be used to evaluate ϕ_f, first order upwinding and central differencing, have inherent problems associated with them. In the case of first order upwinding, solutions that contain strong gradients of ϕ are not accurately predicted because of numerical diffusion. Central differencing on the other hand suffers from numerical instability when the Reynolds number of a given flow is above a restrictively low threshold. Nevertheless, in the framework of structured grids solutions to these problems have been found by using schemes that utilise a second or third order upwind approach to evaluating ϕ_f. Therefore it is sensible to adopt these ideas when deriving a suitable scheme in an unstructured framework. However, because structured second and third order upwind schemes use two node points upstream of a face to evaluate ϕ_f, the lack of line structure in an unstructured grid presents a problem in that no second upstream node exists, see [17].

An obvious solution to this problem is to create an imaginary node on the boundary of the upstream cell as shown in Figure 3.3. A value of ϕ at the imaginary node can be interpolated from the neighbouring nodes and this value used in the functions for ϕ_f already developed for structured grids. Initially this method was adopted here but it proved a complicated approach to solving the problem despite providing accurate results.

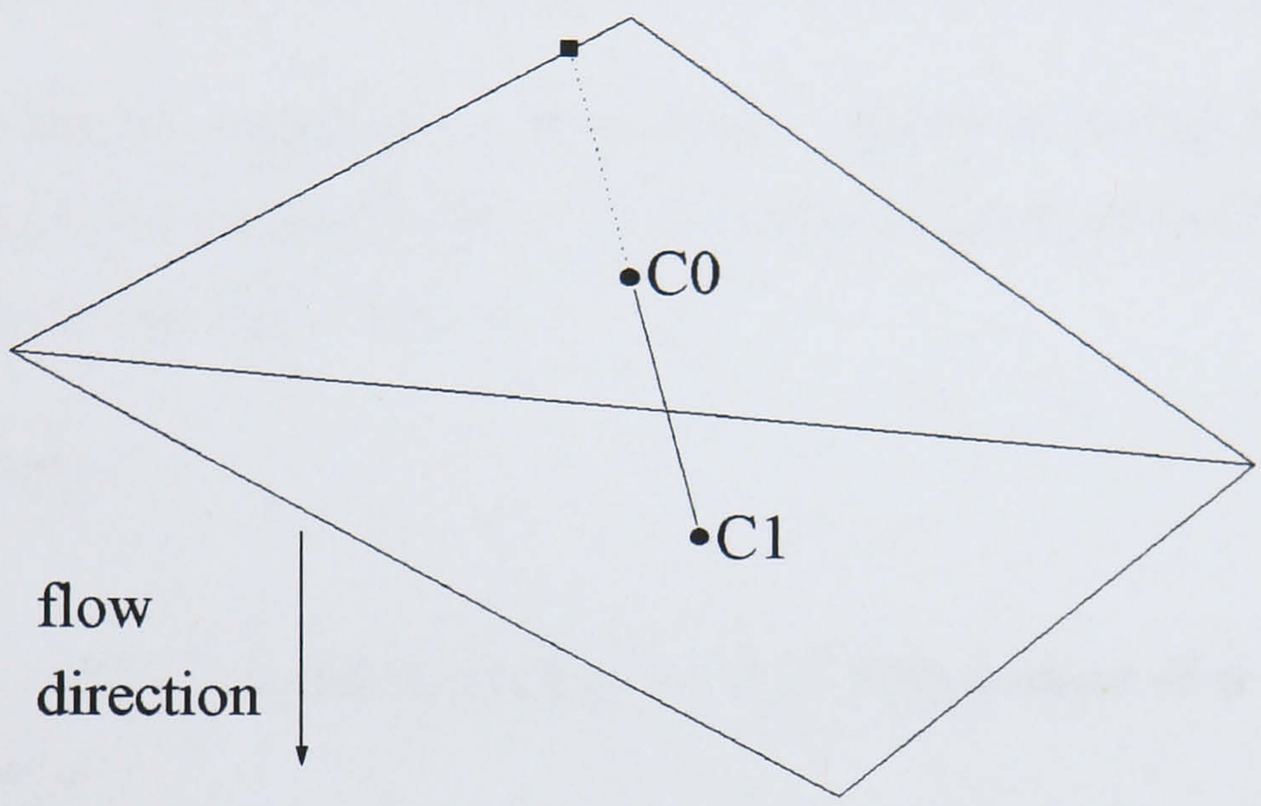


Figure 3.3: Three Node Stencil for the Application of Structured Convective Models to Unstructured Grids. Solid square denotes an imaginary node.

A more elegant solution can be developed if we consider the multidimensional Taylor series expansion of $\phi(\mathbf{x})$ about the centroid of C0 i.e.

Error! Objects cannot be created from editing field codes. (3.20)

where $\Delta\mathbf{x}$ is the vector shown in Figure 3.2 joining the node of C0 to the centre of f, the superscript T represents transpose and $\mathbf{H}_0$ is the Hessian matrix which in two dimensions is defined as

$$\mathbf{H}_0 = \begin{bmatrix} \dfrac{\partial^2 \phi_0}{\partial x^2} & \dfrac{\partial^2 \phi_0}{\partial x \partial y} \\ \dfrac{\partial^2 \phi_0}{\partial y \partial x} & \dfrac{\partial^2 \phi_0}{\partial y^2} \end{bmatrix} \qquad (3.21)$$

Analogous expressions to the second and third order upwind functions used on structured grids can now be created in an unstructured framework by truncating

series (3.20) at an appropriate point. If only the first term in the series is used then ϕ_f is represented by a first order upwind approximation, assuming m_f is positive. By including the gradient term in the series a second order model or linear reconstruction of ϕ_f will be achieved. Third order accuracy or quadratic reconstruction of ϕ_f will result from the inclusion of the Hessian matrix term. However, to include these extra terms in the series the gradient of ϕ and the Hessian matrix must be computed from values of ϕ at the nodes of C0 and its neighbours.

Mathur and Murthy[5] suggested an approximate expression for the gradient term derived from the divergence theorem. The derivation begins from a different form of the divergence theorem to (3.14) i.e.

$$\int_A \phi \mathbf{n} dA = \int_V \nabla \phi dV \qquad (3.22)$$

Discretising (3.22) in a similar manner to (3.15) the gradient of ϕ in C0 can be approximated by

$$\nabla \phi_0 = \frac{1}{\Delta V_0} \sum_f \phi_f \mathbf{A}_f \qquad (3.23)$$

where ϕ_f is linearly interpolated from the nodes of the two cells adjacent to *f*. Consequently $\nabla \phi_0$ is expressed purely in terms of the values of ϕ at the nodes of C0 and its neighbours as required.

Due to the success of third order schemes on structured grids extending an unstructured model to third order accuracy is highly desirable. However, computation of the Hessian matrix cannot be achieved by a simple application of the divergence theorem because mixed derivatives are involved. Instead, a method of finding the Hessian matrix originally used by Delanaye and Essers[69] in an explicit time-marching algorithm has been adopted.

Consider the stencil of cells surrounding C0, pointed to by LISTC and LISTS. In Figure 3.1 for cell 1 this stencil would consist of cells 2 to10. If it is assumed that ϕ behaves as a quadratic function between the nodes of each of the cells belonging to

the stencil, then, from a Taylor series expansion like that given in (3.20) the following system of equations must be satisfied

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 & \frac{1}{2}\Delta x_1^2 & \frac{1}{2}\Delta y_1^2 & \Delta x_1 \Delta y_1 \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ \Delta x_1 & . & . & . & \Delta x_n \Delta y_n \end{bmatrix} \begin{bmatrix} \frac{\partial \phi_0}{\partial x} \\ \frac{\partial \phi_0}{\partial y} \\ \frac{\partial^2 \phi_0}{\partial x^2} \\ \frac{\partial^2 \phi_0}{\partial y^2} \\ \frac{\partial^2 \phi_0}{\partial x \partial y} \end{bmatrix} = \begin{bmatrix} \Delta \phi_1 \\ . \\ . \\ . \\ \Delta \phi_n \end{bmatrix} \qquad (3.24)$$

where n is the number of cells in the stencil surrounding C0, $\Delta \mathbf{x}_n = \mathbf{x}_n - \mathbf{x}_0$ and $\Delta \phi_n = \phi_n - \phi_0$. Clearly, there will not in general be an exact solution to this system of equations. Nevertheless, a solution can be sought by using a least squares technique[70] to arrive at approximate values for the first and second derivatives. Using this technique $\nabla \phi_0$ and $\mathbf{H}_0$ can be expressed in terms of values ϕ at the nodes of C0 and its neighbours.

By using the approximate expressions derived for $\nabla \phi_0$ and $\mathbf{H}_0$ a second or a third order model can be created that is not complicated by the use of imaginary nodes. Unfortunately as in structured meshes the use of higher order schemes has its drawbacks in the form of unbounded solutions. In order to combat this problem a model must have some form of limiter to ensure a solution remains bounded even in the presence of sharp gradients of ϕ. As highlighted in Chapter 2 structured meshes solve this problem by using composite schemes. Whilst a limiter for an unstructured model works on the same principles as a structured scheme its implementation is slightly different. Rather than using different schemes to enforce the monotonicity principle, a function is introduced into the convective model to limit the influence of the high order terms and thereby ensure monotonicity. This form of unstructured limiting has almost exclusively been developed for explicit time-marching codes. However, this does not prevent the same ideas being employed in a pressure based algorithm because the principles from which limiters are developed are general.

Consider a limited form of the unstructured upwind model developed so far

$$\phi_f = \phi_0 + L_0 C_0 \quad L_0 \in [0,1] \tag{3.25}$$

where $C_0=\Delta\mathbf{x}^T\nabla\phi_0$ or $\Delta\mathbf{x}^T\nabla\phi_0+1/2\Delta\mathbf{x}^T\mathbf{H}_0\Delta\mathbf{x}^T$ for a second or third order model respectively and L_0 is the function introduced to limit the high order terms. Barth and Jesperson[71] proposed that L_0 should take the largest admissible value while invoking the monotonicity principle that values of the reconstructed function for C0 must not exceed the maximum and minimum of the node values of C0 and its neighbours. The value of L_0 that satisfies this principle is given by

$$L_0 = \min(L_1, L_2, ..., L_f, ..., L_n) \tag{3.26}$$

where n is the total number faces of C0 and L_f is defined as

$$L_f = \begin{cases} \min\left(1, \phi_0^{\max} - \phi_0 / \phi_f - \phi_0\right) & \text{if } \phi_f - \phi_0 > 0 \\ \min\left(1, \phi_0^{\min} - \phi_0 / \phi_f - \phi_0\right) & \text{if } \phi_f - \phi_0 < 0 \\ 1 & \text{if } \phi_f - \phi_0 = 0 \end{cases} \tag{3.27}$$

ϕ_f in (3.27) is determined by using (3.25) with L_0=1 and $\phi_0^{\min}$ and $\phi_0^{\max}$ are defined as

$$\phi_0^{\max} = \max(\phi_0, \phi_1, ..., \phi_n) \tag{3.28}$$

$$\phi_0^{\min} = \min(\phi_0, \phi_1, ..., \phi_n) \tag{3.29}$$

where n is the number of cells in the stencil surrounding C0.

Whilst the limiter proposed by Barth and Jespersen[71] solved the problem of unbounded solutions it was shown by Venkatakrishnan[72] to suffer from two serious problems. Firstly the limiter was active in near-constant regions of the solution where it was not required and therefore reduced the accuracy of the convective model used. Also, the non-smooth nature of the function L_f caused convergence difficulties when solving the general differential equation (3.9). Venkatakrishnan[72] proposed a new form of the function L_f that would solve these problems given by

$$L_f = \begin{cases} g\left(\phi_0^{\max} - \phi_0 / \phi_f - \phi_0\right) & \text{if } \phi_f - \phi_0 > 0 \\ g\left(\phi_0^{\min} - \phi_0 / \phi_f - \phi_0\right) & \text{if } \phi_f - \phi_0 < 0 \\ 1 & \text{if } \phi_f - \phi_0 = 0 \end{cases} \qquad (3.30)$$

where the function g is defined as

$$g\left(\frac{\Delta_+}{\Delta_-}\right) = \frac{1}{\Delta_-}\left[\frac{\left(\Delta_+^2 + \delta^2\right)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_-\Delta_+ + \delta^2}\right] \qquad (3.31)$$

This new function L_f is smooth and has an additional parameter δ^2 introduced to it that controls the limiter in near constant regions of the solution. δ^2 is taken to be $(T\Delta)^3$ where Δ is an average grid size and T is a constant. To explain how δ^2 works without loss of generality assume that $\Delta_+ = \phi_f - \phi_0$ and $\Delta_- = \phi_0^{\max} - \phi_0$ since other cases can be handled in a similar manner. In near-constant regions of the solution δ^2 will dominate Δ_+^2, Δ_-^2, Δ_+, and Δ_- and therefore function g will reduce to unity i.e. no limiting will be applied in these regions. Where the solution is not near-constant Δ_+^2, Δ_-^2, Δ_+, and Δ_- will dominate δ^2 as Δ_+, $\Delta_- \sim \Delta$ and function g will perform as if δ^2 was zero i.e. limiting appropriately. Clearly, T sets a threshold at which the limiter will be applied and by varying it, more or less limiting can be introduced to the solution. Therefore when a solution is very smooth a high value of T can be applied so that the full benefits of high order accuracy can be realised. Whereas in the case of solutions which have large and rapid changes in ϕ a lower value of T can be used to ensure a bounded solution. There is, therefore, a trade off between high accuracy and bounded solutions.

In summary a bounded, accurate and stable upwind convective model has been developed for use within an unstructured framework. For a face f the model is defined as

$$m_f \phi_f = \begin{cases} m_f\left(\phi_0 + L_0 C_0\right) & \text{if } m_f > 0 \\ m_f\left(\phi_1 + L_1 C_1\right) & \text{if } m_f < 0 \end{cases} \qquad (3.32)$$

or

$$m_f \phi_f = (\phi_0 + L_0 C_0)\langle m_f, 0\rangle - (\phi_1 + L_1 C_1)\langle -m_f, 0\rangle \qquad (3.33)$$

where $\langle A,B\rangle$ denotes the greater of A and B

3.4.2 Diffusion Term

In a similar manner to the convection term we would like to express the diffusion term D_f in terms of the values of ϕ at the nodes of C0 and its neighbours. To determine the form of this term, consider face f in Figure 3.2. The ξ direction is aligned with the vector joining the nodes of C0 and C1 and the η direction is aligned with the vector forming the face itself. We may write

$$\nabla\phi.\mathbf{A}_f = \phi_\xi(\xi_x A_x + \xi_y A_y) + \phi_\eta(\eta_x A_x + \eta_y A_y) \qquad (3.34)$$

where A_x and A_y are the cartesian components of $\mathbf{A}_f$. Expressing the transformation metrics in terms of derivatives of x and y, we have

$$\nabla\phi.\mathbf{A}_f = \phi_\xi\left(\frac{y_\eta A_x - x_\eta A_y}{x_\xi y_\eta - x_\eta y_\xi}\right) + \phi_\eta\left(\frac{-y_\xi A_x + x_\xi A_y}{x_\xi y_\eta - x_\eta y_\xi}\right) \qquad (3.35)$$

For the grid shown in Figure 3.2 consistent approximations for the derivatives can be written as follows

$$\phi_\xi = \frac{\phi_1 - \phi_0}{|\mathbf{e}_f|} \qquad (3.36)$$

$$x_\xi = \frac{x_1 - x_0}{|\mathbf{e}_f|} \qquad (3.37)$$

$$y_\xi = \frac{y_1 - y_0}{|\mathbf{e}_f|} \qquad (3.38)$$

$$\phi_\eta = \frac{\phi_b - \phi_a}{|\mathbf{A}_f|} \qquad (3.39)$$

$$x_\eta = \frac{x_b - x_a}{\left|\mathbf{A}_f\right|} \tag{3.40}$$

$$y_\eta = \frac{y_b - y_a}{\left|\mathbf{A}_f\right|} \tag{3.41}$$

Substituting equations (3.36) to (3.41)into equation (3.35) we obtain

$$\begin{aligned}\nabla\phi.\mathbf{A}_f = &\frac{(\phi_1 - \phi_0)\left[(y_b - y_a)A_x - (x_b - x_a)A_y\right]}{(x_1 - x_0)(y_b - y_a) - (x_b - x_a)(y_1 - y_0)} \\ &+ \frac{(\phi_b - \phi_a)\left[-(y_1 - y_0)A_x + (x_1 - x_0)A_y\right]}{(x_1 - x_0)(y_b - y_a) - (x_b - x_a)(y_1 - y_0)}\end{aligned} \tag{3.42}$$

From the geometry in Figure 3.2 it is clear that

$$A_x = y_b - y_a \tag{3.43}$$

$$A_y = x_a - x_b \tag{3.44}$$

We also note that the vector joining the nodes of C0 and C1 is

$$\mathbf{e}_f = (x_1 - x_0)\mathbf{i} + (y_1 - y_0)\mathbf{j} \tag{3.45}$$

and the vector tangential to face f has components

$$\mathbf{e}_t = (x_b - x_a)\mathbf{i} + (y_b - y_a)\mathbf{j} \tag{3.46}$$

Substituting equations (3.43) to (3.46) into equation (3.42) we may write the diffusion term on face f as

$$\nabla\phi.\mathbf{A}_f = (\phi_1 - \phi_0)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f - (\phi_b - \phi_a)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \mathbf{e}_t.\mathbf{e}_f \tag{3.47}$$

The previous analysis is identical to that used for non-orthogonal structured meshes and is equivalent to a second order central differencing representation of D_f. The first term on the right hand side of equation (3.47) represents the primary gradient whilst the second term is the secondary or cross-diffusion term. On an orthogonal mesh the

cross-diffusion term will be zero as $\mathbf{e}_f$ and $\mathbf{e}_t$ will be perpendicular and on a very regular mesh the term will be dominated by the primary gradient and can then be safely ignored. In general, however, the cross-diffusion term can be significant and its correct evaluation becomes important. Clearly in order to evaluate the cross-diffusion term ϕ_a and ϕ_b must be computed. Typically these values can be found by averaging the values of ϕ at the nodes neighbouring the vertices a and b. In two dimensions this approach to calculating the diffusion term is perfectly adequate. But, extending this methodology to three-dimensional polyhedra of arbitrary shape is not straightforward, as there are no unique face tangential directions or vertices that can be employed to derive an equivalent expression to equation (3.47). It would be possible on a structured hexahedral mesh but clearly this is not what is required.

Davidson[4] and Mathur and Murthy[5] both arrived at an expression for the diffusion term identical to equation (3.47), although Davidson used a different formulation involving an imaginary control volume around face f. As Davidson's[4] work was all two-dimensional this expression was sufficient. However, to find an expression that could be used in both two and three dimensions Mathur and Murthy[5] suggested a change to equation (3.47). Instead of using face tangents and vertices to find the cross-diffusion term they suggested it could be written as the difference between the total diffusion and the primary component across f. Thus,

$$D_f = \Gamma(\phi_1 - \phi_0)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f + \Gamma\left(\overline{\nabla\phi}.\mathbf{A} - \overline{\nabla\phi}.\mathbf{e}\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \qquad (3.48)$$

where the overbar indicates linear interpolation of $\nabla\phi$ from C0 and C1 to the face f. $\nabla\phi$ can be calculated using (3.24) which will result in the diffusion term being expressed solely in terms of values of ϕ at the nodes of C0 and it neighbours as required. Since equation (3.48) is applicable to arbitrary polyhedra that have either two or three dimensions it offers greater flexibility and hence will be adopted here.

3.4.3 Source Term

To complete the discretisation of the general differential equation the source term S_0 must be assigned a representative value for the source in the cell C0. In the case of

the time averaged and instantaneous momentum equation S=-$\partial P/\partial x_i$ and -$\partial p/\partial x_i$ respectively. An approximate value for these source terms in C0 can be calculated using equation (3.24) with ϕ substituted with P or p. The k and ε transport equations take the source terms G_k-$\rho\varepsilon$ or $C_1G_k\varepsilon/k$-$C_2\rho\varepsilon^2/k$ respectively. Clearly these expressions are functions of the dependent variables of the governing equations they belong to. Therefore it is sensible to express this dependence in the discretisation of the source term. Patankar[15] suggested that approximating this dependence linearly would be appropriate i.e.

$$S_0 = S_C + S_G\phi_0 \qquad (3.49)$$

In order to apply sensible values to the constant and gradient terms of S_0, S_C and S_G respectively, we must consider that physically realistic values of k and ε are always positive. Consequently, if in the course of a practical computation negative values of k and ε are calculated then this would have a devastating effect on the rest of the calculation. As Patankar[15] showed this is easily prevented by ensuring that S_C and S_G are positive and negative respectively. This can be achieved by evaluating each term in G_k-$\rho\varepsilon$ and $C_1G_k\varepsilon/k$-$C_2\rho\varepsilon^2/k$ individually and assigning it to S_C or S_G depending on its sign. Obviously the terms assigned to S_G must first be divided by ϕ_0 whilst those assigned to S_C remain as they are.

The individual terms in the k and ε source terms that contribute to S_C and S_G are evaluated in the following manner. The generation term G_k is formed from gradients of the mean velocity which can be approximated in cell C0 using equation (3.24). The values of k and ε at the node of C0 are assumed to prevail throughout the cell and are therefore used to approximate k and ε in the source terms. Density is a constant, as the flow is incompressible, and hence all the expressions in the two source terms are known. In order to keep the discretisation consistent the time averaged and instantaneous momentum equation source terms are cast in the same form as (3.49). Therefore for these source terms S_G=0 and S_C=S_0 with S_0 calculated as described above.

3.5 The Discretised General Convection-Diffusion Equation

It is now sensible to draw together all the discretised terms into the final form of the discretised general convection-diffusion equation. We have

$$a_0\phi_0 = \sum_{nb} a_{nb}\phi_{nb} + b_0 \qquad (3.50)$$

where the summation is over the neighbouring cells of C0 and

$$a_{nb} = \langle -m_f, 0 \rangle + \Gamma\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \qquad (3.51)$$

$$b_0 = \langle -m_f, 0 \rangle L_1 C_1 - \langle m_f, 0 \rangle L_0 C_0 + \Gamma\left(\overline{\nabla\phi}.\mathbf{A} - \overline{\nabla\phi}.\mathbf{e}\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f + S_C\Delta V_0 \qquad (3.52)$$

$$a_0 = \sum_{nb} a_{nb} - S_G\Delta V_0 \qquad (3.53)$$

The terms a_{nb}, b_0 and a_0 are all evaluated explicitly whilst ϕ_0 and ϕ_{nb} are evaluated implicitly. To ameliorate convergence discretised equations are normally under-relaxed. Using Van Doormal and Raithby's[21] E-factor formulation we have

$$a_0\phi_0 = \frac{E}{1+E}\left(\sum_{nb} a_{nb}\phi_{nb} + b_0\right) + \frac{1}{1+E}a_0\phi_0^{old} \qquad (3.54)$$

where the superscript *old* indicates evaluated at the previous iteration and E generally takes a value between 1 and 4.

3.6 Discretisation of the Continuity Equation

In vector form the time averaged continuity equation (3.3) can be written as

$$\nabla.(\rho\mathbf{U}) = 0 \qquad (3.55)$$

The discretisation of this equation is identical to the method used for the instantaneous continuity equation and hence (3.55) is used without loss of

generality. In a similar manner to the general convection-diffusion equation applying the divergence theorem, (3.14), to the left hand side of (3.55) gives

$$\sum_f m_f = 0 \qquad (3.56)$$

As the flow variables are stored at the cell nodes a method of interpolation is required to evaluate the mass flow rate at the cell face. As described in Chapter 2 standard forms of interpolation are likely to cause the velocity and pressure fields to de-couple. To avoid this problem Mathur and Murthy[5], and Davidson[4] independently, derived an interpolation scheme based upon an original idea by Rhie and Chow[20]. For the face *f* this interpolation function is

$$m_f = \rho\overline{\mathbf{U}_0}.\mathbf{A}_f - \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(P_1 - P_0 - \overline{\nabla P_0}.\mathbf{e}_f\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \qquad (3.57)$$

where the overbar indicates interpolation from the two adjacent cells to *f* and the values of ∇P_0 are calculated using (3.24). The first term in equation (3.57) is clearly a simple interpolation of the velocity from the two adjacent nodes surrounding face *f*. However, to avoid de-coupling the pressure term that forms part of the first interpolation in (3.57) must be evaluated in a different manner. Therefore, the second term in (3.57) is added to remove the primary interpolated pressure gradient term and replace it with an analogous term to the primary diffusion term in (3.48).

Whilst this interpolation function solves the problem of the velocity and pressure field de-coupling it has a difficulty associated with it. During an iterative solution of the discretised governing equations the interpolated velocity field in the first term of (3.57) will have been under-relaxed. Madjumdar[73] showed that this could lead to a solution dependent on the under-relaxation factor specified. He offered a solution that will be adopted here which relaxes the mass flow rate separately to the individual velocity components i.e.

$$m_f = \frac{E}{1+E}\left[\rho\overline{\mathbf{U}_0^{nr}}.\mathbf{A}_f - \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(P_1 - P_0 - \overline{\nabla P_0}.\mathbf{e}_f\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f\right] + \frac{1}{1+E}m_f^{old} \qquad (3.58)$$

where the superscript *nr* indicates that no relaxation has been applied to velocity field. In this manner the interpolation function can be used without producing results which are dependent on the relaxation factor.

3.7 The Pressure Correction Equation

The time averaged and instantaneous momentum equation, the k equation and the ε equation can all be discretised by using an appropriate ϕ in equation (3.50). These equations are all non-linear and therefore can be solved by iteration. However, there exists a problem in the calculation of the velocity field from the momentum equation, the unknown pressure field. No obvious equation exists for obtaining the pressure but for a given pressure field there is no difficulty in solving the momentum equations. In fact the pressure field is indirectly specified via the continuity equation. When the correct pressure field is used to solve the momentum equations the resulting velocity field will also satisfy the continuity equation. Patankar's[15] SIMPLE algorithm converted this indirect specification into a direct algorithm for the calculation of pressure and this solution method will be adopted here.

Consider a guessed mean pressure field P^*. A solution of the discretised time averaged momentum equation based on P^* will give an approximate mean velocity field $\mathbf{U}^*$. Again this method can be applied to the instantaneous momentum equation in an identical manner. Using (3.57) this will result in an approximate mass flow rate given by

$$m_f^* = \rho\overline{\mathbf{U}_0^*}.\mathbf{A}_f - \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(P_1^* - P_0^* - \overline{\nabla P_0^*}.\mathbf{e}_f\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \qquad (3.59)$$

Let us propose that the correct mean pressure, mean velocity and mass flow rate are obtained from

$$P = P^* + P' \tag{3.60}$$

$$\mathbf{U} = \mathbf{U}^* + \mathbf{U}' \tag{3.61}$$

$$m_f = m_f^* + m_f' \tag{3.62}$$

where P', $\mathbf{U}'$ and m'_f are the mean pressure, mean velocity and mass flow rate corrections respectively. If we subtract (3.59) from (3.57) we arrive at

$$m_f' = \rho\overline{\mathbf{U}_0'}.\mathbf{A}_f - \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(P_1' - P_0' - \overline{\nabla P_0'}.\mathbf{e}_f\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \tag{3.63}$$

Following Patankar[15] we drop the first term and the interpolated pressure gradient term to arrive at a mass flow rate correction formula i.e.

$$m_f = m_f^* - \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(P_1' - P_0'\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \tag{3.64}$$

This shows how the approximate mass flow rate is corrected in response to the pressure correction and allows a discretised equation for the pressure correction to be derived. Substituting (3.64) into the discretised continuity equation (3.56) gives the pressure correction equation

$$a_0' p_0' = \sum_{nb} a_{nb}' p_{nb}' + b_0' \tag{3.65}$$

where

$$a_{nb}' = \rho\overline{\left(\frac{\Delta V_0}{a_0}\right)}\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_f \tag{3.66}$$

$$b_0' = -\sum_f m_f^* \tag{3.67}$$

$$a_0' = \sum_{nb} a_{nb} \tag{3.68}$$

It can be seen that b_0' in the pressure correction equation is essentially the left hand side of the discretised continuity equation (3.56) evaluated in terms of the approximate mass flow rate. Clearly if b_0' is zero then the guessed pressure field is correct and no pressure correction is required. Therefore b_0' represents a net mass imbalance that the pressure corrections through their association with the velocity corrections must be removed by.

A solution of the pressure correction equation will not give the exact pressure when added to guessed pressure field because of the terms dropped in (3.64). However, the new pressure will be closer to the exact solution than before, and by using this as a new guess for P^* an iterative procedure can be adopted that will gradually find the exact solution. Therefore, after a solution of the pressure correction equation is found the mass flow rate is updated via (3.64) whilst the pressure is updated using the following expression

$$P_0 = P_0^* + \alpha_P P_0' \qquad (3.69)$$

Here, α_P is an under-relaxation factor introduced to help convergence and typically takes values between 0.1 and 0.5.

A velocity correction formula is derived in a similar manner to equation (3.64). Consider, without loss of generality, the approximate discretised time averaged x-momentum equation

$$a_0 U_0^* = \sum_{nb} a_{nb} U_0^* + b_0^* \qquad (3.70)$$

subtracting (3.70) from the exact form of this discretised equation we have

$$a_0 U_0' = \sum_{nb} a_{nb} U_0' + b_0' \qquad (3.71)$$

dropping the summation term and those terms in b_0' relating to the diffusion and convective terms of the governing differential equation we arrive at the mean x-velocity correction formula i.e.

$$U_0 = U_0^* - \frac{\mathbf{i}.\nabla P_0'}{a_0} \qquad (3.72)$$

where $\nabla P_0'$ is calculated using equation (3.24). Combining all three velocity components into a general vector form we have

$$\mathbf{U}_0 = \mathbf{U}_0^* - \frac{\nabla P_0'}{a_0} \qquad (3.73)$$

3.8 Solution Algorithm

The solution algorithm can now be summarised by the following steps

1. Guess initial velocity, pressure k and ε fields.
2. Solve the discretised momentum equations.
3. Calculate the mass imbalance for each cell.
4. Solve the pressure correction equation.
5. Correct the pressure and velocity fields.
6. Solve the discretised k and ε equations.
7. Treat the corrected pressure, velocity k and ε as a new initial guess and return to step 2.
8. Repeat steps 2-7 until convergence is achieved.

The pressure correction equation is solved using an adapted form of algebraic multigrid for unstructured meshes, which will be described in detail in Chapter 4. The convergence criterion for the P' equation is that the normalised residual defined as

$$R_{P'} = \frac{\sum_{cells}\left(\left|\sum_{nb} a'_{nb}P'_{nb} + b'_0 - a'_0P'_0\right|\right)}{\sum_{cells}\left(\left|\sum_{nb} a'_{nb}P'_{nb} + b'_0 - a'_0P'_0\right|\right)_{ref}} \qquad (3.74)$$

has reduced to less than 0.25. In equation (3.74) the numerator is the sum of the absolute values of the residuals of all the cells in the domain. This is divided by a reference total residual, which is calculated after one iteration of the algebraic multigrid.

No convergence criterion is set for the solution of the discretised momentum, k and ε governing equations during an individual global iteration. Instead, the Gauss-Seidel point by point method used to solve these equations is restricted to only perform one sweep of the computational domain for each equation. However, global normalised residuals are defined for the continuity, momentum and k and ε equations to allow a global convergence criterion to be specified. These are defined as

$$R_c = \frac{\sum_{cells}\left(\left|\sum_{f} m_f^*\right|\right)}{\sum_{cells}\left(\left|\sum_{f} m_f^*\right|\right)_{ref}} \qquad (3.75)$$

$$R_\phi = \frac{\sum_{cells}\left(\left|\sum_{nb} a_{nb}\phi_{nb} + b_0 - a_0\phi_0\right|\right)}{\sum_{cells}\left|a_0\phi_0\right|} \qquad (3.76)$$

In the normalised residual of the continuity equation, R_c, the reference residual is taken as the largest total residual in the first five iterations.

3.9 Boundary Conditions

At the boundaries of the computational domain special conditions must be applied to the discretised equations to reflect the physical characteristics of the boundary. Boundaries take a number of forms such as wall, inlet and periodic boundaries to name only a small selection. Here, only the conditions applied to the most common

boundaries will be discussed. However, there are numerous other types of boundaries that will require different treatment, examples of which can be found in the open literature.

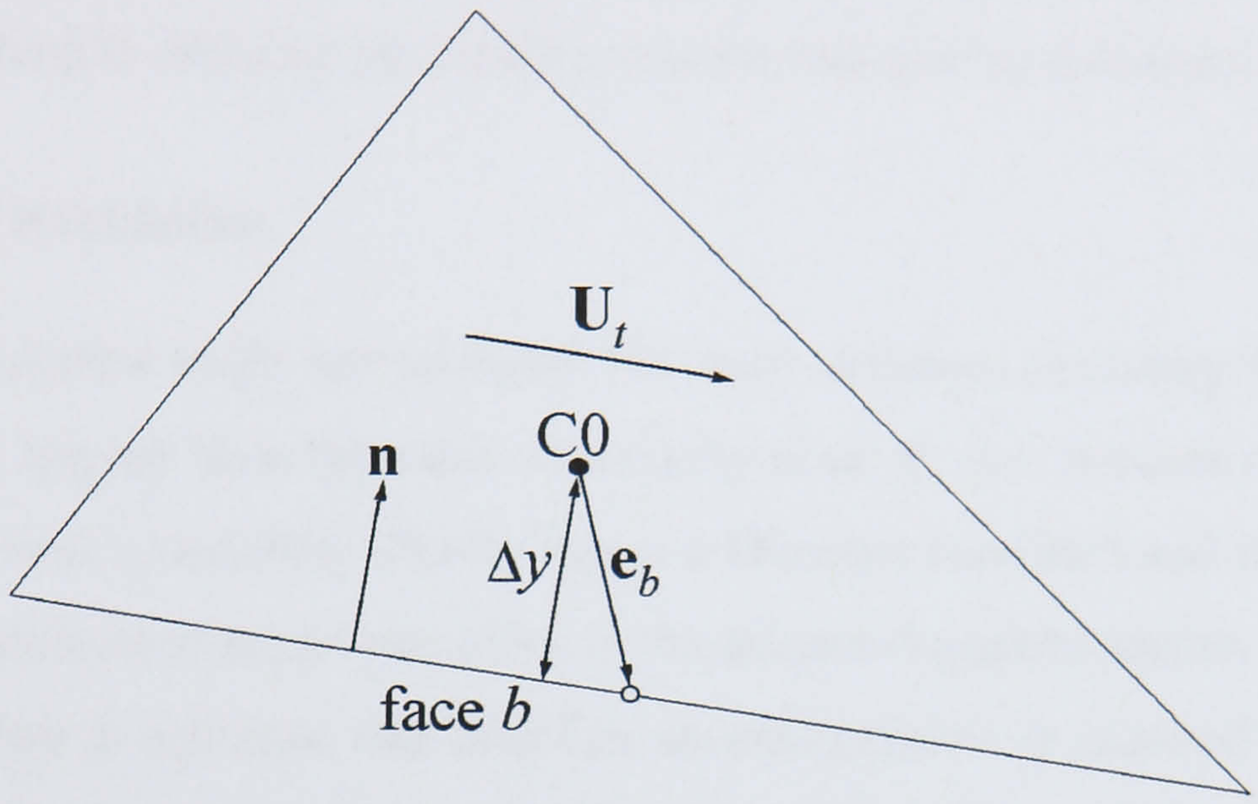


Figure 3.4: Boundary Control Volume Schematic

Whilst different boundary conditions are applied to the various governing equations in different ways the general convection-diffusion equation effectively has only two general types of boundary condition applied to it. These two conditions are the Neumann and Dirichlet boundary condition and their application to equation (3.50) is dealt with here. On a Neumann boundary the gradient of ϕ normal to the boundary is specified. This replaces the approximate value calculated for the diffusion term, using (3.48), in the discretised equation. The convective term is calculated from the cell adjacent to the boundary as if it was the upstream cell on an interior face. Alterations are required to the calculation of the source term, but they are specific to the particular type of physical boundary represented by the Neumann condition, and will be dealt with later. The Dirichlet boundary condition specifies values of ϕ on the boundary. The diffusive flux is approximated using an expression suggested by Mathur and Murthy[5]. For the boundary face in Figure 3.4 this is given by

$$D_b = \mu(\phi_b - \phi_0)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}_b}\right)_b + \mu\left(\nabla\phi_0.\mathbf{A} - \nabla\phi_0.\mathbf{e}_b\,\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}_b}\right) \qquad (3.77)$$

where $\mathbf{e}_b$ is the vector in Figure 3.4 that joins the cell node to the centre of the boundary face b and ϕ_b is the specified boundary variable. As in (3.48) the first term

in (3.77) is treated implicitly and the second term explicitly. The specified value at the boundary replaces the approximate value calculated using the convective model (3.33) and again adjustments to the source term will be dealt with later. Having described the implementation of these two general boundary conditions the specifics of applying them to different physical boundaries will now be discussed.

3.9.1 Wall Boundaries

Rigid impermeable walls are probably the most common boundary to fluid flow problems. In laminar flow the value of velocity is set to zero because of the no-slip condition at wall boundaries. Clearly this is a Dirichlet condition and is imposed on the instantaneous momentum equations in the manner described earlier. The pressure at the boundary is unknown and therefore an extrapolation is required in order that the source term can be calculated with equation (3.24). The method adopted here is to use a first order extrapolation i.e. set the pressure at the boundary equal to the adjacent node pressure. As the velocity is known at the boundary the mass flow rate will also be specified and hence no mass flow rate correction will be required. This conveniently removes the need for values of P' to be known outside of the computational domain. If the Poisson-like nature of the pressure correction equation is considered this is equivalent to applying a Neumann boundary condition to the P' equation.

In the case of turbulent flows the implementation of wall boundary conditions needs special treatment. This is due to the existence of turbulent boundary layers, across which steep variations of flow properties occur. In these circumstances the standard high Reynolds number k-ε turbulence model becomes inadequate and a different approach must be adopted. One method is to employ a very fine grid near the wall and use a low Reynolds number form of the k-ε turbulence model[52]. This method is computationally expensive and an alternative and popular approach is to use wall functions[52] to bridge the turbulent boundary layer region. The wall functions are developed from a one-dimensional Couette flow analysis, which is assumed to be valid near to the wall. Peric[74] presented a general method of applying these functions to all forms of grid and it is his approach that will be adopted here.

The aim of a wall function is to replace the calculated diffusive flux in the discretised time averaged momentum equation, with a value that more realistically represents the velocity gradient between the boundary face and its adjacent node. This is achieved by assuming that the node adjacent to the boundary face is outside of the boundary layer, and that an empirical relation can model the velocity profile between the node and the face. Consider then the wall shear stress, τ_w, expressed as a function of the adjacent nodal velocity component parallel to the wall, $\mathbf{U}_t$, see Figure 3.4. For simplicity it is assumed that τ_w acts in the direction opposite to $\mathbf{U}_t$ i.e.

$$\tau_w = -\lambda_w \mathbf{U}_t \qquad (3.78)$$

where the coefficient λ_w is determined from a two part universal velocity profile expression[75] given by

$$\lambda_w = \begin{cases} \dfrac{\mu}{\Delta y} & \text{if } y^+ < 11.225 \\ \dfrac{\rho C_\mu^{\frac{1}{4}} k_0^{\frac{1}{2}} \kappa}{\ln\left(E y^+\right)} & \text{if } y^+ > 11.225 \end{cases} \qquad (3.79)$$

where Δy is the normal distance from the boundary face to the adjacent node shown in Figure 3.4 and y^+ is calculated from

$$y^+ = \frac{\rho C_\mu^{\frac{1}{4}} k_0^{\frac{1}{2}} \Delta y}{\mu} \qquad (3.80)$$

In the equations above κ is von Karman's constant and is taken as 0.42. E is an empirical constant, which depends on wall roughness and through which other effects such as pressure gradients and mass transfer can be accounted for[52]. In the case of smooth impermeable walls E is assigned the value 9.81.

Since the discretised time averaged momentum equation is resolved in the directions of the Cartesian co-ordinates, the resultant shear wall force, $\mathbf{T}_w$, that will be used to replace the calculated diffusive flux, also needs to be expressed in terms of its

Cartesian components. This force is equal to the product of the wall shear stress τ_w and the area of the boundary face $|\mathbf{A}_b|$ i.e.

$$\mathbf{T}_w = \tau_w |\mathbf{A}_b| = -\lambda_w |\mathbf{A}_b| \mathbf{U}_t \qquad (3.81)$$

The components of $\mathbf{T}_w$ can be readily obtained if the tangential velocity, $\mathbf{U}_t$, can be expressed in terms of its Cartesian components. This can be achieved by subtracting from the total velocity, **U**, its component in the direction normal to the wall. This component is calculated by taking the scalar product of **U** and the unit normal vector, **n**, shown in Figure 3.4. By performing this analysis we arrive at the following expression for the shear wall force

$$\mathbf{T}_w = -\lambda_w |\mathbf{A}_b| [\mathbf{U} - (\mathbf{U}.\mathbf{n})\mathbf{n}] \qquad (3.82)$$

In incompressible flow the shear wall force is equivalent to the three components of the diffusive flux in the discretised momentum equations. Rather than replacing the diffusive flux on the wall boundary directly it is more convenient to introduce an imaginary velocity on the wall, $\mathbf{U}_w$. When this velocity is used in the diffusive flux calculation at the boundary the resulting flux is equal to the value given by equation (3.82). Some simple algebraic manipulations of equation (3.77) give an appropriate expression for $\mathbf{U}_w$ as

$$\mathbf{U}_w = \mathbf{U}_0 + \frac{\mathbf{A}.\mathbf{e}_b}{\mathbf{A}.\mathbf{A}} \left(\frac{\mathbf{T}_w}{\mu} - \nabla \mathbf{U}_0 . \mathbf{A}_b \right) + \nabla \mathbf{U}_0 . \mathbf{e}_b \qquad (3.83)$$

The convective and source terms of the time averaged momentum equation are handled in an identical manner to the method described earlier for laminar flow, as is the pressure correction equation. However, the equations for k and ε also require special treatment in cells near to the wall. In the k equation the diffusive term is set to zero[52] as is the convective term because the mass flow rate through the wall will be zero. Modifications to the source term are also required and will be described below.

Within the turbulent boundary layer a balance between production and dissipation of turbulent kinetic energy is assumed[50] giving the following relation for ε

$$\varepsilon = \frac{C_{\mu}^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa \Delta y} \qquad (3.84)$$

The generation of turbulent kinetic energy, G_k, in the turbulent boundary layer can be approximated, under Couette flow assumptions, by

$$G_k = |\tau_w| \frac{\partial |\mathbf{U}_t|}{\partial \mathbf{n}} = |\tau_w| \frac{|\mathbf{U}_t|}{\Delta y} \qquad (3.85)$$

Using equations (3.84) and (3.85) the k equation source term can be approximated the following relation

$$S_k = |\tau_w| \frac{|\mathbf{U}_t|}{\Delta y} - \rho \frac{C_{\mu}^{\frac{3}{4}} k^{\frac{3}{2}}}{\kappa \Delta y} \qquad (3.86)$$

The transport equation for ε is abandoned for the cells adjacent to wall boundaries due to its inapplicability there. Instead, ε is fixed to the value given by equation (3.84).

3.9.2 Inlet Boundaries

At the inlet to a computational domain the velocity is specified. In both laminar and turbulent flow this Dirichlet boundary condition is applied to the discretised momentum equation in the manner described earlier. The pressure at the inlet, required to calculate the momentum equation source term, is found using the first order interpolation as it was on wall boundaries. As before if the velocity at the boundary is known then the mass flow rate can be calculated and hence a Neumann boundary condition is applied to the pressure correction equation.

In turbulent flow values of k and ε must also be specified at the inlet to the computational domain. This can be achieved either by using values obtained from experimental data or by using empirical formulae. It is generally the case that k and ε are unknown and therefore the most common approach is to use empirical formulae. k is normally estimated as a percentage of the square of the inlet velocity i.e.

$$k = \frac{3}{2}(IU_{in})^2 \qquad (3.87)$$

where I is the intensity of turbulence whose value depends on the particular fluid flow under investigation and U_{in} is the magnitude of the inlet velocity. The dissipation of turbulent kinetic energy can then be computed from equation (2.58) i.e.

$$\varepsilon = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{l} \qquad (3.88)$$

where the turbulent length, l, is determined by the geometry of a given flow domain. For a duct flow l will be typically taken as the hydraulic diameter. As in the momentum equation the specified values k and ε are Dirichlet boundary conditions and are applied to the appropriate discretised equations in the manner already described.

3.9.3 Outlet Boundaries

Two possible boundary conditions can be applied to the outflow from a computational domain. The first of these is known as the derivative boundary condition and can only applied to domains that have just one outlet. The condition states that the normal derivative to the boundary of all variables is equal to zero i.e.

$$(\nabla\phi.\mathbf{n})_b = 0 \qquad (3.89)$$

This Neumann boundary condition is imposed on the discretised momentum, k and ε equations using the procedure described above. However, applying this condition to the pressure correction equation requires special consideration.

Previously, Neumann boundary conditions have been applied to the P' equation when the mass flow rate at a boundary has been specified. Here no mass flow rate has been specified and therefore it must be calculated. On other boundaries surrounding a computational domain the mass flow rate will be known and hence from the principle of mass conservation the mass flow rate at the outlet can be estimated. In fact, if the P' equation is surrounded on all sides by Neumann

boundaries, satisfying mass conservation in the computational domain is essential if the pressure correction is to be guaranteed to converge[76].

The method by which the outlet mass flow rate is estimated is as follows, calculate an approximation of the outlet mass flow rate using the following adjusted form of (3.64) suggested by Mathur and Murthy[77]

$$m_b = \rho \mathbf{A}_b.\mathbf{V}_0 - \frac{\rho \Delta V_0}{a_0}\left(P_b - P_0 - \nabla P_0.\mathbf{e_b}\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_b \qquad (3.90)$$

Then find the sum of b'_0 for all the cells in the computational domain, which will give the net mass imbalance. From this the outlet mass flow rate can be adjusted so that mass is conserved by dividing the net imbalance between each of the outlet faces in an area-weighted manner. By applying Neumann boundary conditions to all the boundaries surrounding a computational domain solutions of the pressure correction equation will also be non-unique. This is not a desirable situation and therefore to ensure uniqueness one node value in the domain is set equal to zero.

One disadvantage of using the derivative boundary condition is that it can only be used when there is only one outflow. For cases where there is more than one outflow specifying pressure is a more appropriate condition. For unstructured meshes this boundary condition also offers other considerable benefits for cases with just one outflow. These will be discussed in Chapter 5 with some illustrative examples. Specifying the pressure at outlet only effects the way in which the P' equation boundary conditions are implemented. The momentum and k and ε equations still have a Neumann boundary condition applied to them in the same way as the derivative condition. If the pressure is known at a boundary then clearly there is no need to apply a correction i.e. $P_b'=0$. This condition is included in the pressure correction equation by applying the following analysis. Consider the boundary mass flow correction formula below

$$m_b = m_b^* - \rho \frac{\Delta V_0}{a_0}\left(P_b' - P_0'\right)\left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}}\right)_b \qquad (3.91)$$

This formula is derived from equation (3.90) in an analogous way to the derivation of equation (3.64) from equation (3.57). Substituting equations (3.91) and (3.64) into the discretised continuity equation, (3.56), we arrive at the following boundary pressure correction equation

$$a'_0 P'_{nb} = \sum_{nb} a'_{nb} P'_{nb} + a'_b P'_b + b'_0 \qquad (3.92)$$

where the summation is over the interior neighbours to C0, a'_{nb} is given by equation (3.66), and

$$a'_b = \rho \frac{\Delta V_0}{a_0} \left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}} \right)_b \qquad (3.93)$$

$$a'_0 = \sum_{nb} a'_{nb} + a'_b \qquad (3.94)$$

$$b'_0 = -\sum_f m^*_f - m^*_b \qquad (3.95)$$

Clearly the term containing P'_b in equation (3.92) will drop out and leave an expression just in terms of the interior cells as required. Also, with pressure specified along one boundary unique solutions to the pressure correction equation will be found without the need to hold one node value fixed.

3.9.4 Symmetry Boundaries

Many flow patterns are symmetric and this property can be exploited to make a considerable reduction in the grid size required to model a particular problem. A typical example of this is shown in Figure 3.5. This diagram shows the cross section of a square duct and the two lines of symmetry that cut the cross section. Clearly only a quarter of the pipe needs to be modelled, as the other three quadrants will all contain identical solutions. The appropriate boundary condition to enforce this principle is that the velocity normal to boundary is zero and that the normal derivative of **U**, k and ε is also zero i.e.

$$(\mathbf{U}.\mathbf{n})_b = 0 \qquad (3.96)$$

$$(\nabla\phi.\mathbf{n})_b = 0 \qquad (3.97)$$

Clearly equation (3.96) and (3.97) imply that the convective and diffusive fluxes in the momentum, k and ε equations will be zero at the boundary and as the mass flow rate is specified a Neumann boundary condition will be applied to the P' equation.

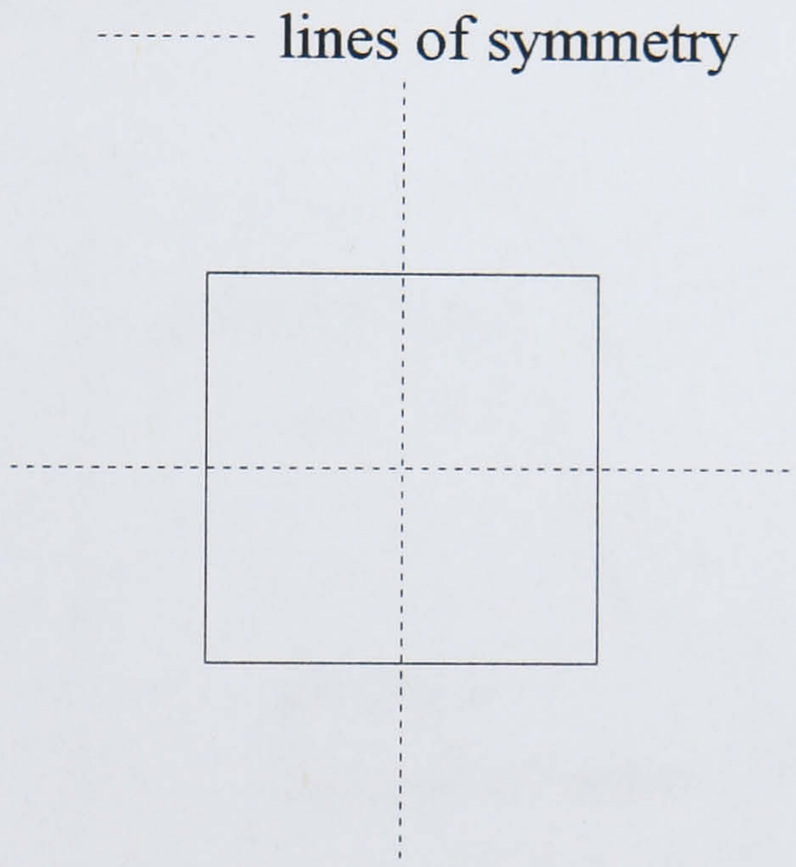


Figure 3.5: Symmetry Boundaries in a Square Duct Cross-Section

3.9.5 Periodic Boundaries

Another type of symmetry boundary is the periodic boundary. This form of boundary is commonly found in turbomachinery applications and will be explained with an example from this field. Consider the fluid flowing between two blades of a rotor stage in a turbine, it is reasonable to assume that the flow properties of this fluid will be identical to the flow between each of the blades of the stator. In other words the flow pattern has a periodically repeating nature. It is therefore sensible only to model the flow between two of the blades. To illustrate this point consider the two-dimensional representation of the passage between two stator blades given in Figure 3.6. This figure shows two sets of boundaries labelled periodic and periodic-shadow. Along these boundaries it is assumed that all variables satisfy the following condition

$$\phi(x, y) = \phi(x, y + L) \qquad (3.98)$$

To implement this condition in the governing equations the faces on the periodic and periodic-shadow boundaries must be grouped into pairs of corresponding faces, an example of which is shown in Figure 3.6. The grouped pairs held in LISTP, described in section 3.3.6, are then used to adjust LISTC such that the cells adjacent to the grouped pairs are treated as if they were geometric neighbours on the grid. In this way the periodic boundaries can be treated as any other interior face and hence avoid any further complication.

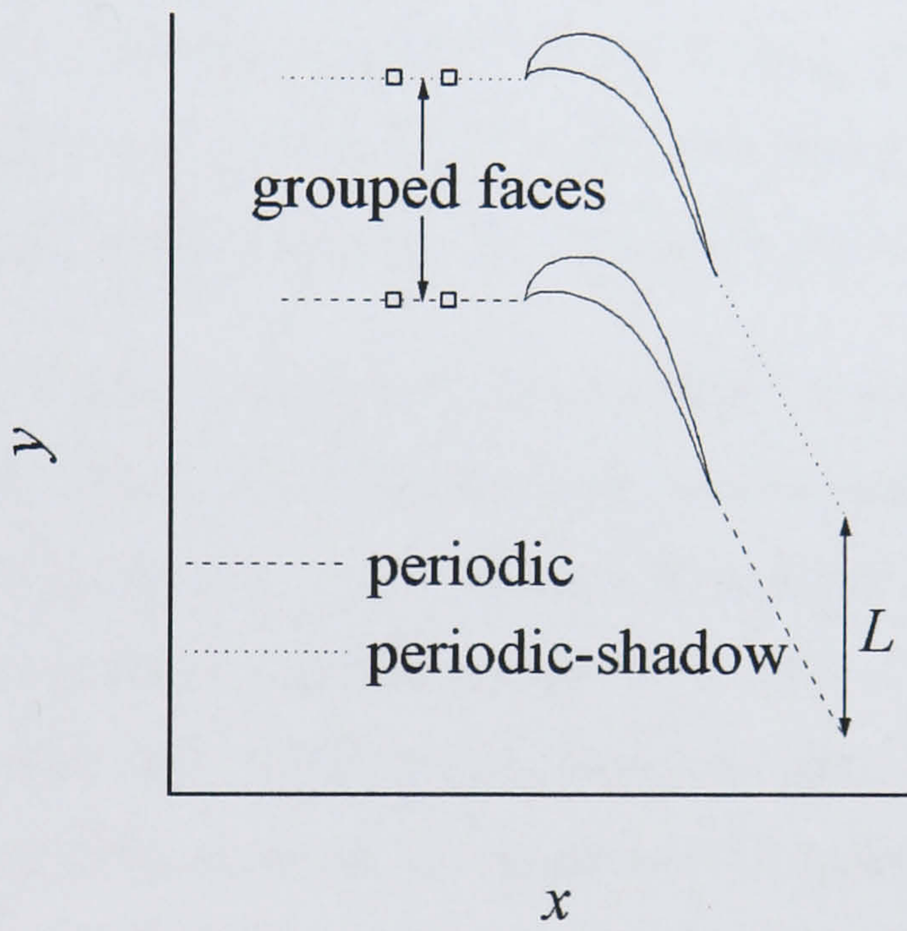


Figure 3.6: Periodic Boundaries for 2D Rotor Blades. Open squares denote vertices.

3.10 Conclusions

A finite volume discretisation strategy has been detailed that is appropriate for use with non-orthogonal unstructured grids. The scheme can be used to discretise both the laminar and turbulent governing equations and includes an accurate, stable and bounded convective model. An adapted form of the SIMPLE algorithm for unstructured meshes has been presented as the solution method and the implementation of the most common boundary conditions described. The thesis will now go on to describe a solution method for the pressure correction equation (3.65) in Chapter 4 and then investigate thoroughly the accuracy of the method described in this chapter, in Chapters 5 and 6.

CHAPTER 4

4 Algebraic Multigrid

4.1 Introduction

The pressure correction equation (3.65) derived in Chapter 3 must be solved to a satisfactory accuracy, typically the normalised residual, R_p, reduced to less than 0.25, for every iteration of the SIMPLE algorithm. Therefore, for any solution method based on SIMPLE, an efficient method of solving (3.65) is essential for the algorithm to solve the complete set of governing equations in a reasonable time frame.

As highlighted in Chapter 3 the simple Gauss-Seidel[78] point by point method used to solve the discretised momentum, k and ε equations is not appropriate for the solution of the pressure correction equation. This is because of an innate failing of the Gauss-Seidel method that causes the number iterations required to solve a given discretised equation to increase in a worse than linear scaling with grid size. To alleviate this problem SIMPLE solution methods based on structured meshes typically used efficient iterative linear solvers such as the line by line method[15], Stone's method[79] or incomplete Cholesky conjugate gradients[80]. However, since these rely on their being some structure in the mesh they are not practical for use with unstructured meshes. This leaves a limited choice of suitable solution algorithms for solving the pressure correction equation on unstructured grids and typically the methods are little better than the poor performing Gauss-Seidel method. Nevertheless, it is possible to accelerate the convergence of the Gauss-Seidel method through the use of multigrid.

Multigrid schemes rely upon using a hierarchy of grids, from fine to coarse, to solve a discrete set of equations using a given solution method. Each grid in the hierarchy is particularly effective at removing errors of wavelength characteristic of the grid spacing associated with that mesh. In this way the errors in a solution are reduced much more rapidly thereby reducing the number of iterations to achieve a convergent solution. In

addition to this the multigrid method uses less computational effort on the coarser meshes yielding an additional efficiency gain.

The hierarchy of grids used in multigrid methods is often chosen *a priori*[81], but this method relies on a known mesh structure and therefore once again is impractical when applied to unstructured grids. A solution to this problem is offered by algebraic multigrid that creates the coarser meshes from the equation matrix itself and therefore can be applied to structured and unstructured meshes alike. Here, an algebraic multigrid scheme is developed for unstructured meshes that is based upon the work of Hutchinson and Raithby[82]. A detailed derivation of the method is presented at the beginning of the chapter with particular attention given to the problems of storing and calculating connectivity data in the hierarchy of meshes. This is followed by presentation of the results from a simple test case to compare calculations using a number of multigrid cycling strategies. Finally conclusions are drawn and the appropriate method to use to solve the pressure correction equation selected

4.2 Derivation of the Additive Correction Equation

Consider the general form of the Poisson equation,

$$\nabla^2 \phi = S \qquad (4.1)$$

This is analogous to the pressure correction equation (3.65) and therefore can be used without loss of generality throughout this chapter to demonstrate the algebraic multigrid strategy.

Following the same method used to discretise the general convection-diffusion equation in Chapter 3, i.e. integrate equation (4.1) over a volume, V, with surface, A, apply the divergence theorem and then approximate the resulting expression on a small volume ΔV_0, we arrive at the following discretised form of the general Poisson equation,

$$a_0 \phi_0 = \sum a_{nb} \phi_{nb} + b_0 \qquad (4.2)$$

where the summation is over the neighbouring cells of C0 and

$$a_{nb} = \left(\frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}} \right)_f \qquad (4.3)$$

$$b_0 = \left(\overline{\nabla \phi}.\mathbf{A} - \overline{\nabla \phi}.\mathbf{e} \frac{\mathbf{A}.\mathbf{A}}{\mathbf{A}.\mathbf{e}} \right)_f - S_0 \Delta V_0 \qquad (4.4)$$

$$a_0 = \sum_{nb} a_{nb} \qquad (4.5)$$

The vectors $\mathbf{A}_f$ and $\mathbf{e}_f$ are shown in Figure 3.2, S_0 is the constant value of the source term in C0 and the overbar indicates linear interpolation of $\nabla \phi$ from C0 and C1 to the face f.

The discretised set of equations (4.2) can be solved using Gauss-Seidel point by point iteration and although inefficient this method forms the basis of the additive correction strategy and therefore it is instructive to describe it here. If an initial guess of the values of ϕ in a given computational domain is given by ${\phi^1}_i$ and ${\phi^k}_i$ are the values obtained from iteration k, where k=2, 3, ..., then the Gauss-Seidel method of solution can be described by the following expression,

$$\phi_0^{k+1} = \frac{\sum_{nb} a_{nb} \phi_{nb}^k + b_0^k}{a_0} \qquad (4.6)$$

where the superscript k on the RHS of the equation represents the most recently calculated value of ϕ for a given location in the computational domain. Convergence of the discretised equations is judged by monitoring how the residual R^k changes with k, defined as

$$R^k = \frac{\sum_{cells} \left| \phi_0^k - \frac{\sum_{nb} a_{nb} \phi_{nb}^k + b_0^k}{a_0} \right|}{R^1} \qquad (4.7)$$

where the summation is over all the cells in the computational domain.

The slow convergence of the Gauss-Seidel method is caused by it's inability to remove low frequency errors in the solution. That is those errors that are typically of the same order of magnitude as the length of the computational domain when Fourier analysis is

applied to the error in a numerical solution. However, the Gauss-Seidel method is efficient in removing errors of high frequency therefore if the damping of low frequency errors could be improved then an efficient solution algorithm would be achieved. Multigrid improves the damping of low frequency errors by using the Gauss-Seidel method to solve the same problem on increasingly coarser meshes. Clearly on a coarser mesh what was a low frequency error will become a high frequency error and therefore the Gauss-Seidel method will be able to remove it more efficiently.

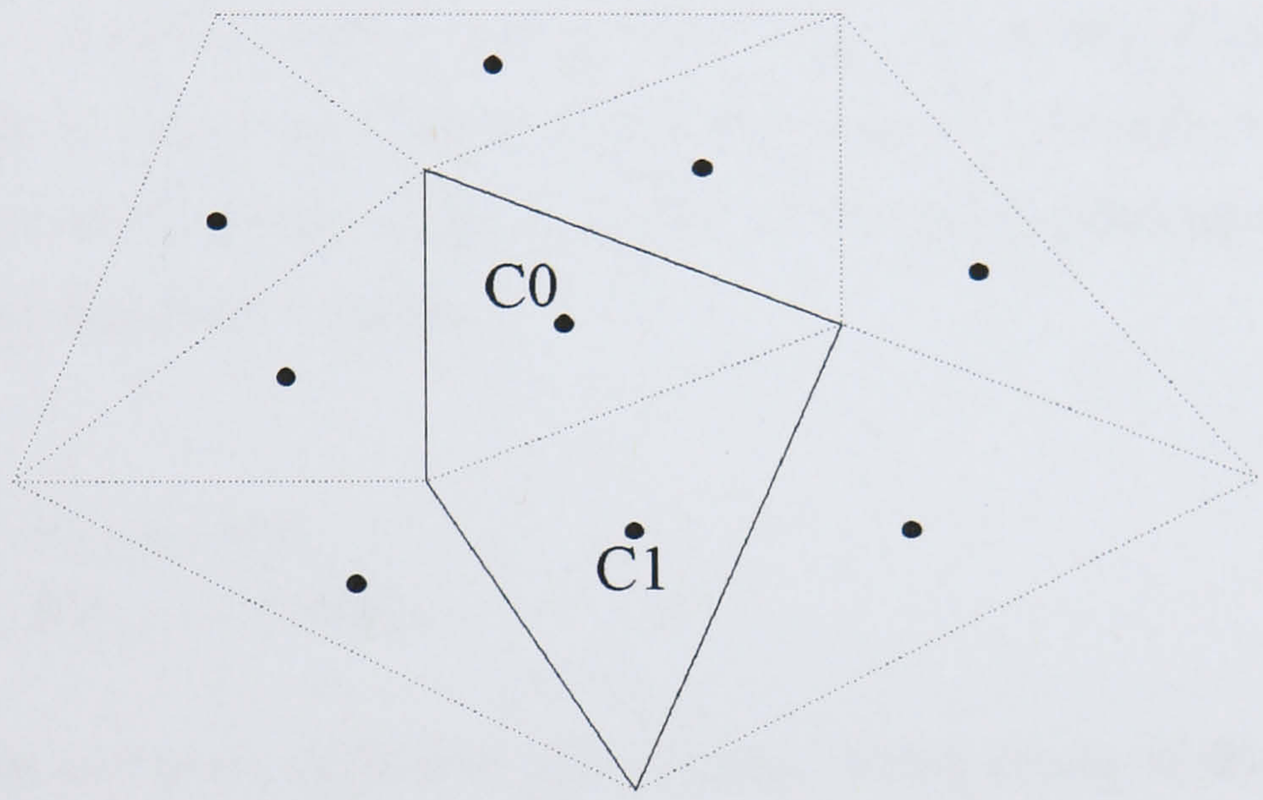


Figure 4.1: Multigrid Control Volume Schematic: Dotted lines represents the underlying fine mesh, solid lines denote the coarse mesh block B0 and solid circles denote fine mesh nodes.

The additive correction method forms the coarse grid equations by asserting integral conservation over blocks of control volumes. It then solves these new equations on the coarser grid and passes back corrections to the finer grid, which ensure integral conservation is satisfied over the blocks of the fine grid, and by doing so removes low frequency errors which stall the convergence. In order to derive these equations consider the grid shown in Figure 4.1. To form the coarse mesh equation block B0 from cells C0 and C1, see Figure 4.1, consider (4.2) rewritten as follows

$$a_0\phi_0 - \sum_{nb} a^*_{nb}\phi_{nb} = \sum_{nb} \hat{a}_{nb}\phi_{nb} + b_0 \qquad (4.8)$$

where

$$a_{nb} = a^*_{nb} + \hat{a}_{nb} \qquad (4.9)$$

and

$$a^*_{nb} = 0 \text{ if } \phi_{nb} \text{ is not in the same coarse mesh block as C0}$$
$$\hat{a}_{nb} = 0 \text{ if } \phi_{nb} \text{ is in the same coarse mesh block as C0} \qquad (4.10)$$

Summing the n equations in the coarse block surrounding C0 we have

$$\sum_n \left(a_0 \phi_0 - \sum_{nb} a^*_{nb} \phi_{nb} \right)_n = \sum_n \left(\sum_{nb} \hat{a}_{nb} \phi_{nb} + b_0 \right)_n \qquad (4.11)$$

If ϕ_0 is the best estimate of ϕ_0 on the fine grid, the correction δ_0 obtained on the coarse grid is added to ϕ_0 and to the values of ϕ in the other cells contained within the B0, in order to obtain an improved solution. Similarly, cells not contained within B0 have corrections added to them depending on which coarse block they belong to. This can be expressed in mathematical terms by

$$\phi_0 = \phi'_0 + \delta_0$$
$$\phi_{nb} = \phi'_{nb} + \delta_0 \text{ if } \phi_{\text{nb}} \text{ is in B0} \qquad (4.12)$$
$$\phi_{nb} = \phi'_{nb} + \delta_{nb} \text{ if } \phi_{\text{nb}} \text{ is not in B0}$$

where δ_{nb} is the correction applied to cells in neighbouring blocks to B0. Substituting (4.12)into (4.11) gives

$$\sum_n \left(a_0 (\phi'_0 + \delta_0) - \sum_{nb} a^*_{nb} (\phi'_{nb} + \delta_0) \right)_n = \sum_n \left(\sum_{nb} \hat{a}_{nb} (\phi'_{nb} + \delta_{nb}) + b_0 \right)_n \qquad (4.13)$$

and rearranging yields the final form of the additive correction equation,

$$\bar{a}_0 \delta_0 = \sum_n \left(\sum_{nb} \hat{a}_{nb} \delta_{nb} \right)_n + \bar{b}_0 \qquad (4.14)$$

where

$$\bar{a}_0 = \sum_n \left(a_0 - \sum_{nb} a^*_{nb} \right)_n \qquad (4.15)$$

and

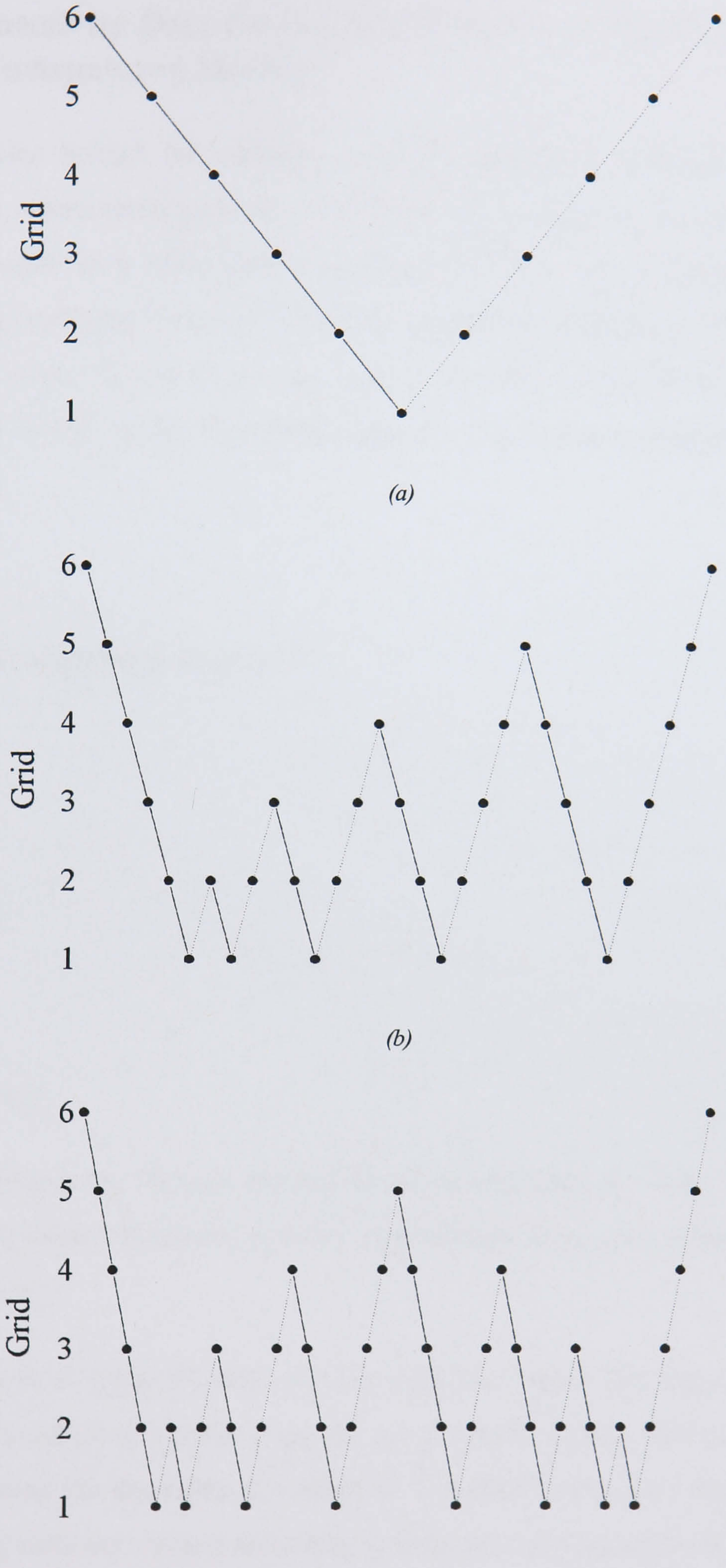
$$\bar{b}_0 = \sum_n \left(\sum_{nb} a_{nb} \phi'_{nb} + b_0 - a_0 \phi'_0 \right) \qquad (4.16)$$

Clearly the source term $\bar{b}_0$ is the sum of the residuals of the fine grid equations for the cells in B0. This is consistent with our principle that the correction δ_0 should improve upon the existing solution ϕ_0, as it is the error in ϕ_0 that drives what value of δ_0 will be calculated from the coarse grid equations.

The additive correction multigrid strategy now becomes clear. Having achieved an approximate solution of the fine grid equations using the Gauss-Seidel point by point method, this is used to form the coarse grid equations. These in turn are approximately solved using Gauss-Seidel and the result used to correct the fine grid solution. The procedure then repeats itself again but now taking the newly corrected solution as the starting point. In this way the errors of wavelength associated with each grid are removed quickly and hence overall convergence is improved.

However, the algorithm can be extended beyond two grids, because the coarse grid itself can be used to create a still coarser mesh and so on until the grids can be coarsened no further. This will obviously yield greater improvements in efficiency because with a whole hierarchy of grids errors of all wavelengths will be removed quickly on their associated grid. Therefore, one cycle of this multigrid strategy consists of moving through the grids from finest to coarsest solving the relevant equations on each grid and creating the next set of coarse equations from these solutions. Then, when the grid cannot be coarsened any further the corrections are passed up through the grids again solving the relevant equations again based on the corrected solution.

The cycle described above is commonly know as the V-cycle and is shown diagrammatically in Figure 4.2. Also shown in Figure 4.2 are two other more complicated strategies, the F-cycle and the W-cycle. These can provide greater convergence rates although extra computational time will clearly be used for a cycle, and therefore a judgement has to made as to which cycle is most appropriate for a given problem. This will be investigated with respect to the solution of the pressure correction equation later in this chapter.



(a)

(b)

(c)

Figure 4.2: Three Multigrid Cycling Strategies (a) V-Cycle (b) F-Cycle (c) W-Cycle, the solid circle denotes an application of the Gauss-Seidel method, the dotted lines indicates the passing of corrections between grids whilst the solid line denotes the passing of residuals.

4.3 Connectivity Data for the Additive Correction Multigrid Strategy on Unstructured Meshes

The principles behind the additive correction multigrid strategy are quite simple, however on unstructured grids the calculation of connectivity data for the hierarchy of grids can result in a considerable overhead in terms of the computational storage requirements and time. Lonsdale[83] offered one solution in the form of a coarsening MxN matrix, **K**, where M and N are the number of cells on the coarse and fine meshes respectively. If the set of discretised equations (4.2) are expressed in the following matrix form

$$\mathbf{A}\phi = \mathbf{b} \qquad (4.17)$$

then the coarse equations are given by

$$\mathbf{A}'\delta = \mathbf{b}' \qquad (4.18)$$

where

$$\mathbf{A}' = \mathbf{K}\mathbf{A}\mathbf{K}^{\mathrm{T}} \qquad (4.19)$$

and

$$\mathbf{b}' = \mathbf{K}(\mathbf{b} - \mathbf{A}\phi) \qquad (4.20)$$

However, despite the obvious attractions of this scheme, for large three-dimensional problems the matrix **K** would become prohibitively large and therefore here, another solution is sort.

The problem is to group the cells of a fine grid into blocks and then calculate a pointer system as described in Chapter 3 for the set of coarse blocks. The grouping of cells is performed using the algorithm described by Lonsdale[83]. Consider the cell C0, first it's neighbouring cells are ordered according to their strongest connections, that is according to the values of the relevant coefficient a_{nb}. Then C0 is grouped with the cell it is most strongly connected. This is repeated for all cells in the fine grid unless either cell under consideration or it's most strongly connected neighbour already belongs to a coarse grid group. Having created a number of groups by iterating once through the set of fine grid

cells a second pass is made. In this pass any fine grid cells which do not already belong to a group are assigned to be a coarse grid cell in their own right unless the coarse grid has become half the size of the fine grid in which case the fine cell is attached to the group that contains their most strongly connected cell. The data created by this algorithm is stored in LISTG, which has the following form,

LISTG(i, 1)=total number fine cells in group i

LISTG(i, 2)=index of fine cell 1

LISTG(i, 2)=index of fine cell 2

.

.

LISTG(i, LISTG(i, 1)+1)=index of fine cell LISTG(i, 1)

Having formed LISTG it is now necessary to form LISTC for the coarse mesh, see Chapter 3 for a definition. To do this the fine cells in a coarse block are looped over and the neighbouring fine cells for each member of the coarse block tested to see if it belongs to the block in question or not. If not, then that neighbour is added to the list of fine cells that are adjacent to the particular coarse block. This is repeated for all of the blocks in the coarse grid. Each of the adjacent fine cells can then be tested to find out which of the coarse blocks it belongs to. Hence, for each coarse block a list of the number of neighbouring coarse blocks and their index can be constructed, in other words LISTC for the coarse mesh. The other connectivity lists presented in Chapter 3, such as LISTCF and LISTF, are not required for coarse meshes because cell face information and boundary conditions have already been included implicitly in the coarse mesh equations when they are created from the original fine mesh.

Clearly, calculating the coarse mesh connectivity data can be a lengthy process particularly for the large grids used in three-dimensional problems. Therefore, when the multigrid process is used as part of the SIMPLE algorithm, to solve the pressure correction equation, the calculation of the connectivity data for the hierarchy of grids is only performed once, during the first global iteration. The connectivity data is stored for

the remaining iterations taking advantage of the fact that each coarse grid is half the size of it's preceding fine grid, to reduce the size of the arrays used to store the data. In this manner whilst the first global iteration takes time to calculate the hierarchy of grids connectivity data, subsequent iterations can access the data immediately without too great a storage penalty.

4.4 Results

In order to test this multigrid strategy the following form of the Poisson equation was solved

$$\nabla^2 \phi + \sin(\pi x)\cos(\pi y) = 0 \qquad (4.21)$$

where

$$-\frac{1}{2} \le x \le \frac{1}{2} \qquad (4.22)$$

and

$$0 \le y \le 1 \qquad (4.23)$$

The Neumann boundary condition,

$$\nabla \phi . \mathbf{n} = 0 \qquad (4.24)$$

where **n** is the outward pointing normal to the boundary, was applied to (4.21). Consequently, in order to obtain a unique solution one node value was set equal to the following unique solution of (4.21)

$$\phi = \frac{1}{2\pi^2}\sin(\pi x)\cos(\pi y) \qquad (4.25)$$

A schematic diagram of the computational domain is shown in Figure 4.3, and Figure 4.4 shows the smaller of the two triangular meshes that were used to calculate solutions to (4.21). The smaller mesh contained 946 CV (control volumes) whilst the larger grid was made up of 2564 CV. Calculations were performed starting from an initial guess of ϕ=0 throughout the computational domain and were iterated for 500 multigrid cycles.

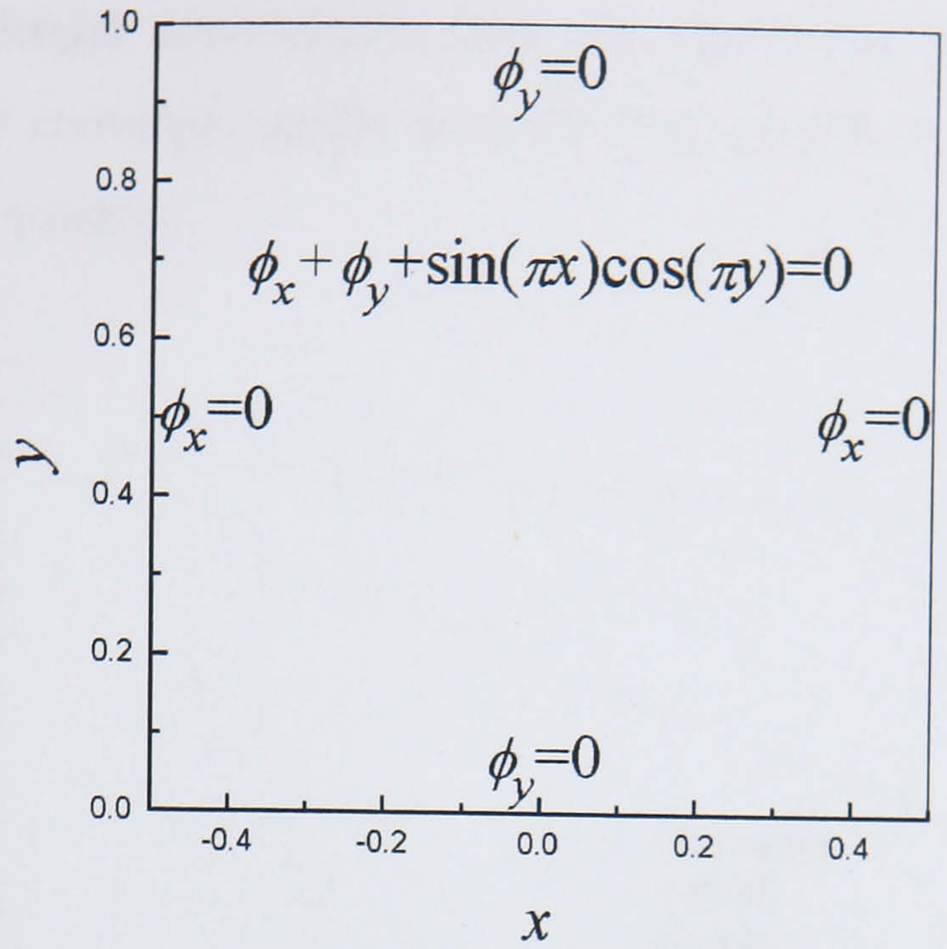


Figure 4.3: Schematic of the Computational Domain used to test the Multigrid Algorithm

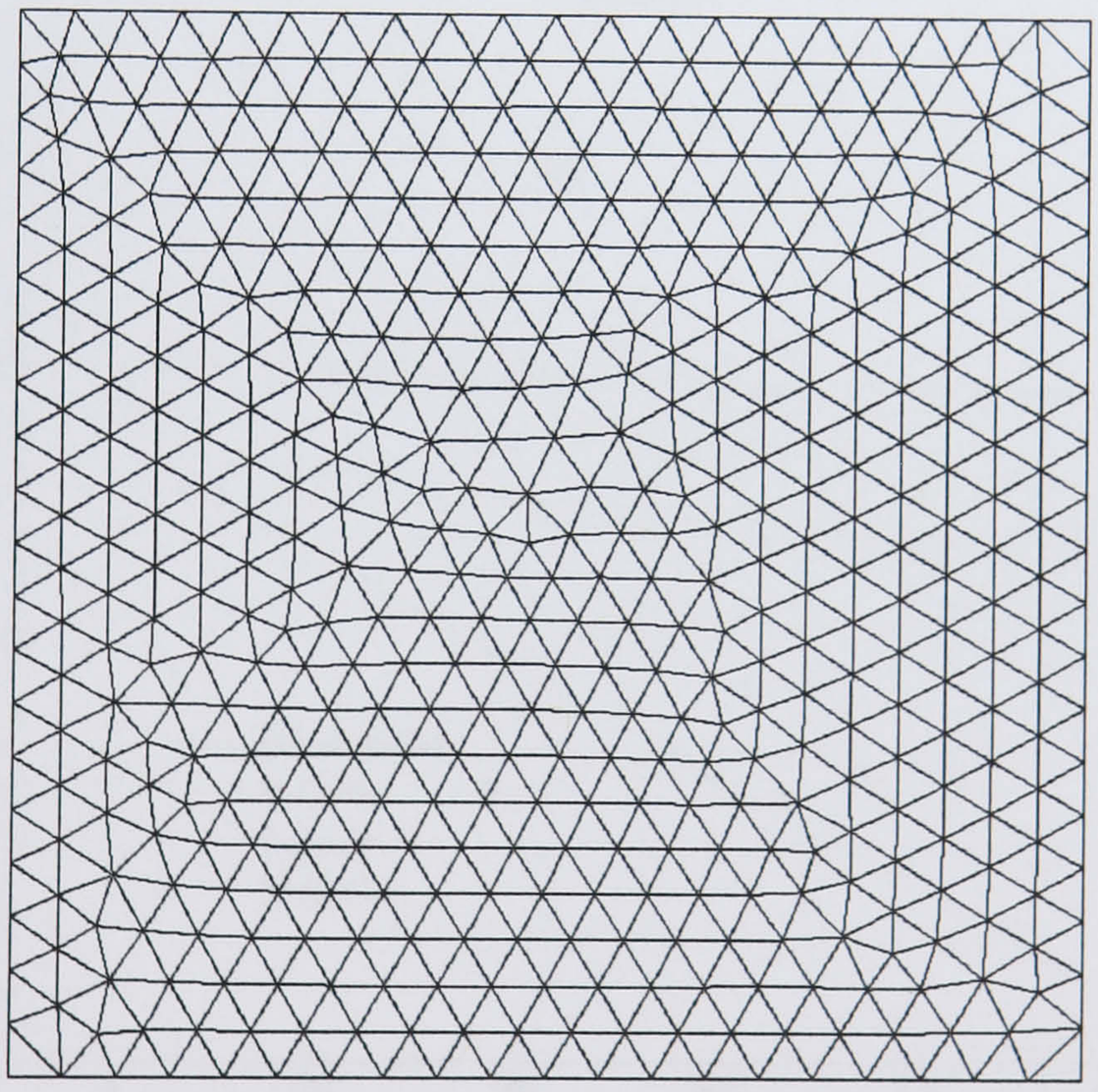
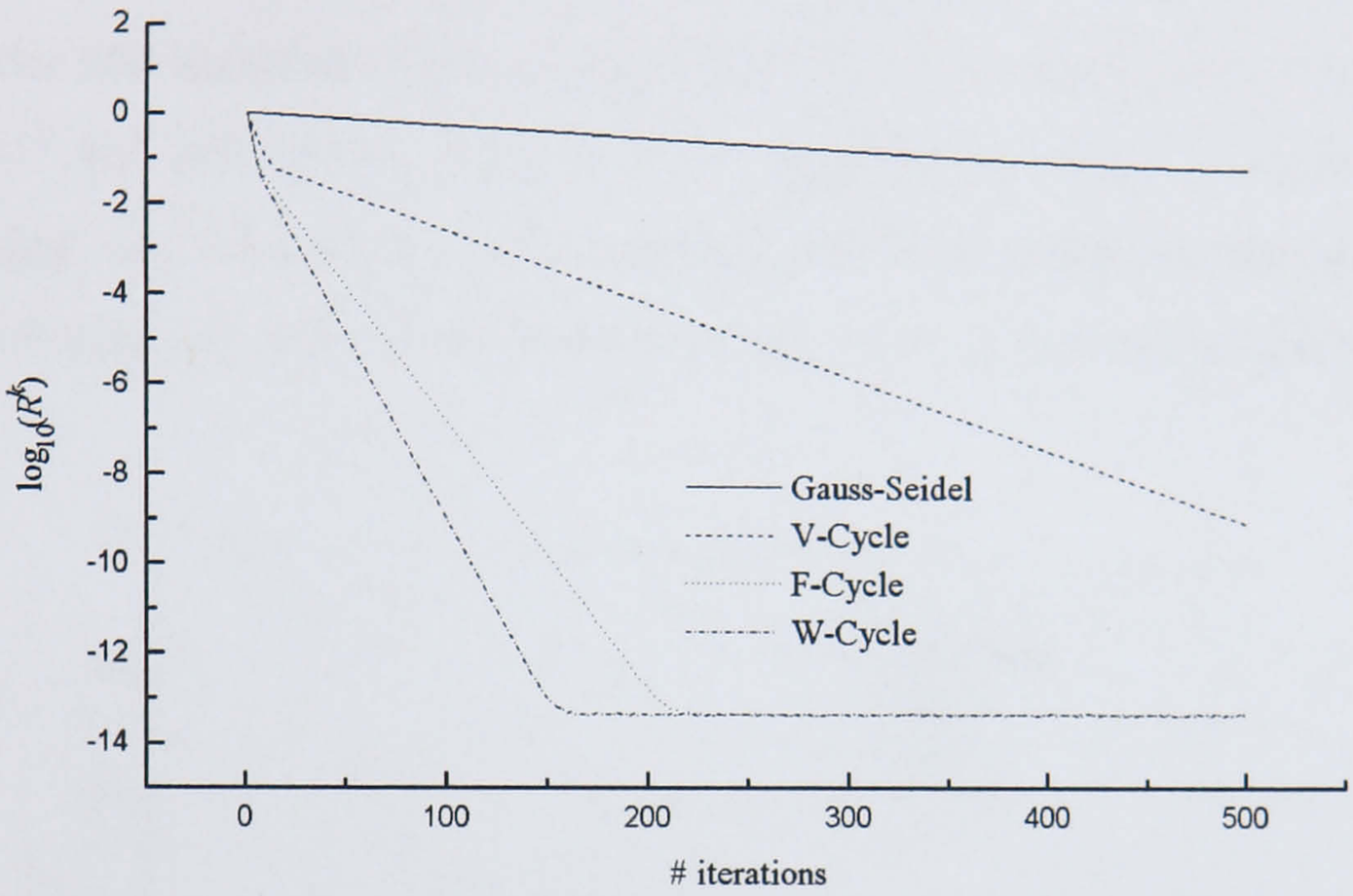


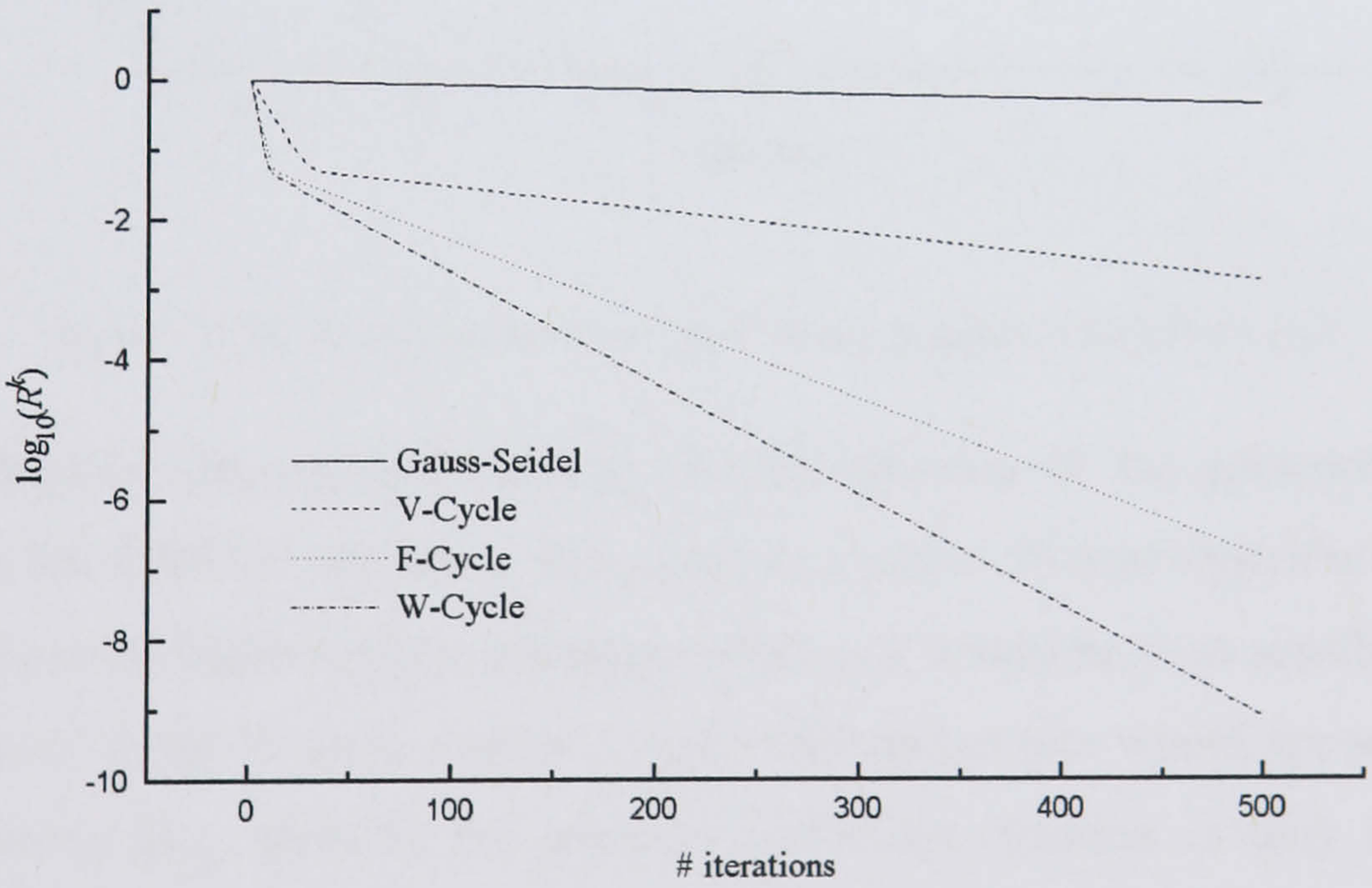
Figure 4.4: 946 CV Triangular Unstructured Mesh used to test Multigrid Algorithm

Solutions were sort using the V, F and W-cycle as well as ordinary Gauss-Seidel point by point iteration by way of a reference point. Figure 4.5 shows the convergence rates of the various cycles for the 946 CV and 2564 CV meshes. A clear pattern is evident, Gauss-Seidel converges very slowly because of it's inability to quickly remove the

errors that have longer wavelengths than the underlying fine grid size. Whilst the multigrid schemes converge rapidly because they are able to remove all the different error wavelengths quickly.



(a)



(b)

Figure 4.5: Convergence Histories for (a) 946 CV (b) 2564CV

Nevertheless, there exist differences in the how the individual multigrid schemes perform. For both meshes the W-cycle has the highest convergence rate followed closely

by the F-cycle and then considerably slower is the V-cycle. It is obvious then that the added complexity of both the W and F cycles gain considerable benefits in terms of convergence rate. Another point to notice about Figure 4.5 is that the convergence rate reduces as the grid size increases. For the 946 CV mesh the W-cycle has hit the machine error in under 200 iterations whilst for the 2564 CV in 500 iterations the error is of the order of 10^{-9} but still falling. This is to be expected but what is important is that increasing grid size does not have the crippling effect that it does on Gauss-Seidel, and therefore solutions can still be sort in large meshes within a reasonable time frame.

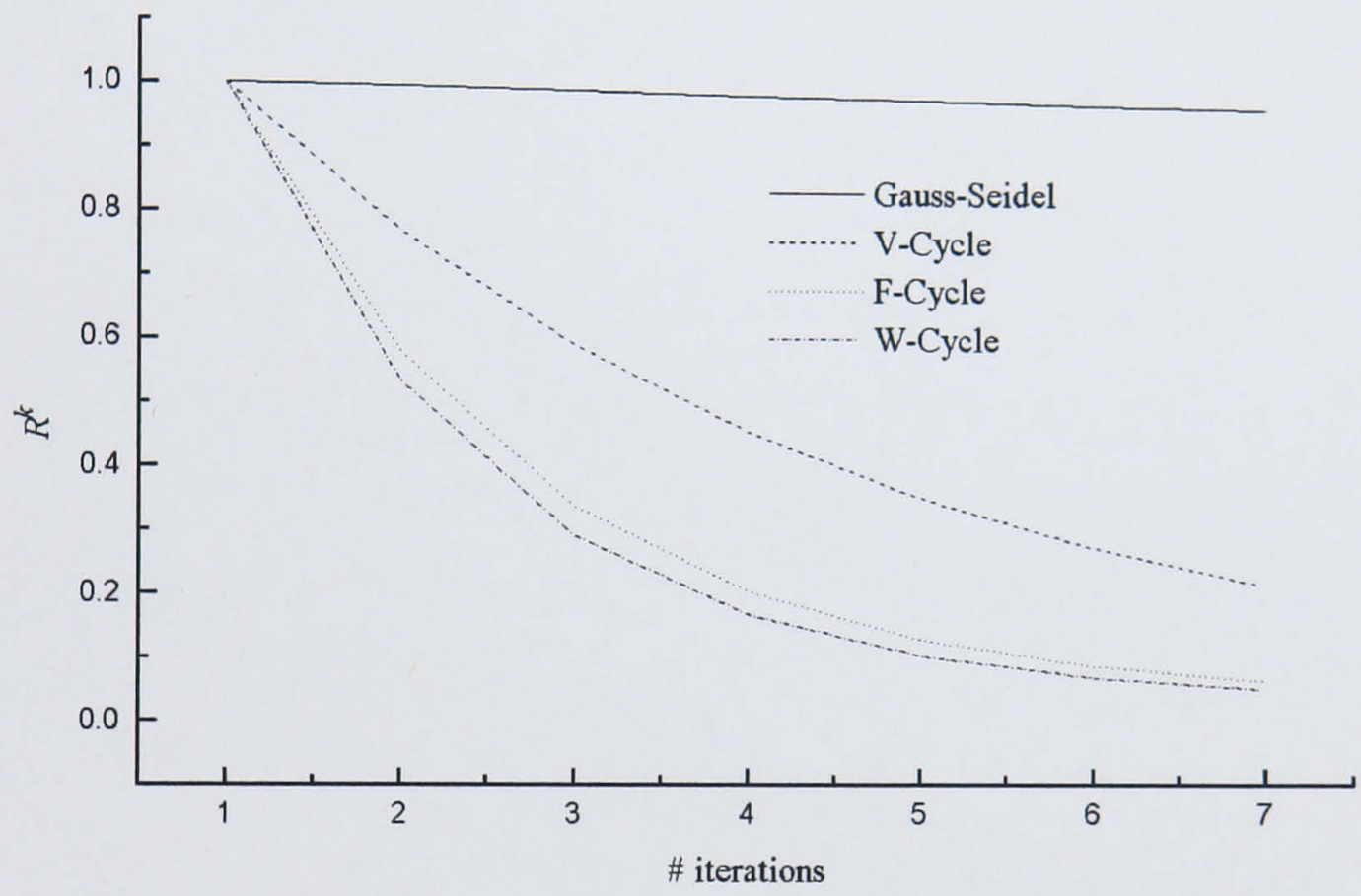


Figure 4.6: The First Seven Iterations of the Various Schemes on the 946 CV grid

Clearly, adopting the multigrid strategy for the solution of the pressure correction equation in the SIMPLE algorithm will result in a more efficient algorithm. However, out of the three multigrid cycling strategies which one would be most sensible to adopt. On the face of it the W-cycle with it's high convergence rate would appear to be the obvious choice. But, typically the pressure correction equation is only solved to a tolerance of $R/R_1 < 0.25$, therefore a closer inspection of the early part of the convergence history of the various cycles is warranted. Figure 4.6 shows the first 7 iterations on the 946 CV mesh. Although again the W and F-cycles have a higher convergence rates, the point at which the normalised residual becomes less than 0.25 is only a matter of 3 or 4 iterations apart. Therefore, given that the V-cycle takes less time to complete one cycle because of it's simplicity there is little to choose between the schemes and consequently the simplest scheme, the V-cycle, has been chosen here.

4.5 Conclusions

A multigrid strategy has been presented here that can be easily applied to equations discretised on unstructured grids. It has been tested on a problem analogous to the solution of the pressure correction equation in the SIMPLE algorithm and shown to be a great improvement on Gauss-Seidel point by point iteration. Three cycling strategies were tested and after considering the specific application of the multigrid scheme to the solution of the pressure correction equation the simplest, the V-cycle, was chosen. This solution method for the pressure correction equation will be adopted throughout this thesis.

CHAPTER 5

5 Laminar Test Cases

5.1 Introduction

The solution method described in Chapters 3 and 4 must be tested to ensure that it accurately predicts the values of the governing equations' dependent variables over a wide range of problems. To begin with, this testing is performed using three benchmark test cases that have been widely used throughout the open literature, and one test case for which an analytic solution can be found. These four test cases are flow in a skew lid driven cavity, flow over a backward facing step, Poiseuille flow between two flat plates and flow in a square duct with strong curvature. The first three of these cases are two-dimensional geometries whilst the fourth is three-dimensional. Each of the test cases are calculated at Reynolds number sufficiently small for the flow to be laminar. This removes the complexity of including the turbulent kinetic energy and dissipation equations in the solution algorithm, and therefore makes analysis of any errors in the results easier to comprehend. However, because the two turbulent transport equations are discretised in an identical manner to the momentum equations, laminar flow calculations are still representative of the accuracy of the complete algorithm.

The results of the calculations for three benchmark test cases i.e. the skew lid driven cavity, the backward facing step and the square duct with strong curvature, are compared with published data. This data is both experimental and numerical, and consequently facilitates a comparison not only with the real flow patterns, but also against other codes that have been developed for the solution of the Navier-Stokes equations. The analytic solution of the flow between two flat plates is also used to validate the code, because values of all the dependent variables can be compared throughout the computational domain. This allows a complete check to be made of

the accuracy of the solution, which is not possible with published experimental or numerical data.

The test cases are chosen to include the most important features of the complex flows in encountered in practical engineering problems. This allows the algorithm to be judged on realistic flow patterns. The first, second and third order convective models described in Chapter 3 are compared to ascertain whether higher order schemes have significant benefits. Also, the effect of grid refinement on the algorithm is considered, as are the convergence characteristics of the different convective schemes. The impact that algebraic multigrid has upon the efficiency of the algorithm is quantified and compared to the computational cost of it's implementation. By choosing a wide variety of test cases the most common boundary conditions are all tested to ensure they are correctly executed and are valid for a given situation. Overall these four laminar test cases provide sufficiently complicated flow patterns to adequately test the basic solution algorithm, and analyse the effects that multigrid and higher order convective modelling have upon the accuracy and efficiency of the method.

5.2 Flow in a Skew Lid Driven Cavity

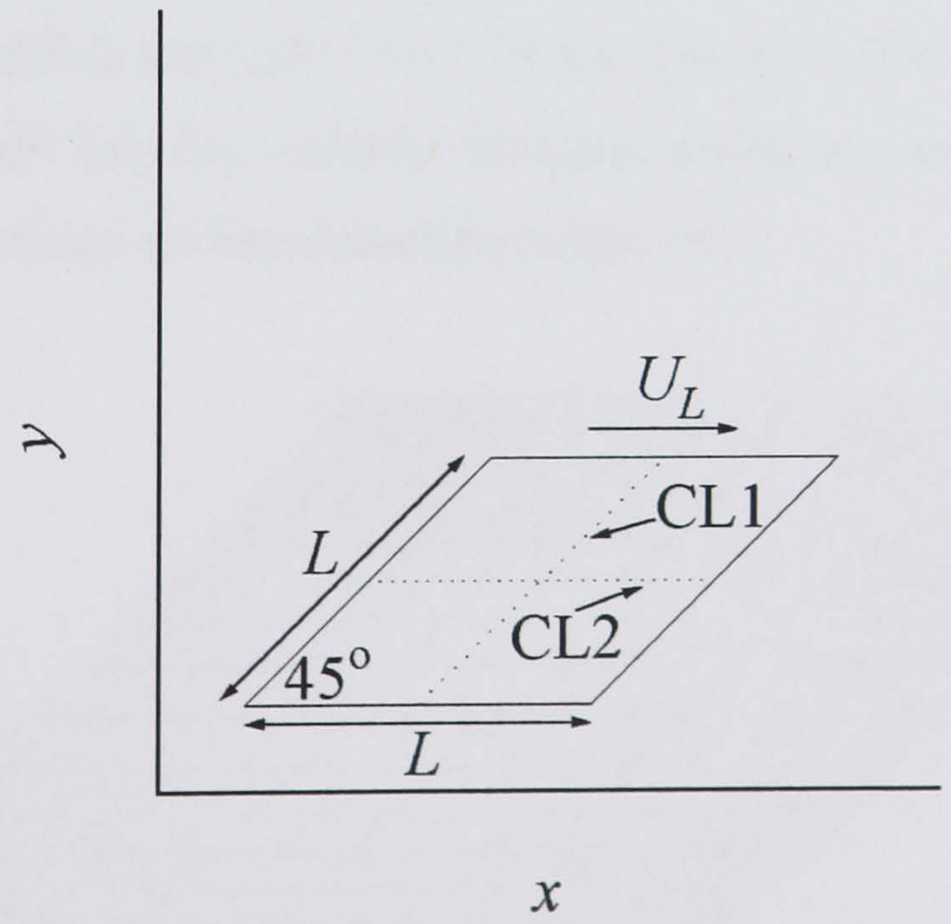


Figure 5.1: Schematic of the Skew Lid Driven Cavity

The lid driven cavity flow has long served as a standard test case for incompressible Navier Stokes solvers[84]. A recirculating flow is generated in a cavity by moving a lid across the top of the fluid at constant velocity. Classically the cavity was square in shape and a structured orthogonal grid could easily mesh the computational domain. However with the advent of solvers intended for use with structured non-orthogonal meshes, a cavity formed from a parallelogram has become a more appropriate test case. A schematic diagram of the geometry of the parallelogram or skew cavity is shown in Figure 5.1.

Demirdzic et al.[85] and Oosterlee et al.[86] have both published benchmark solutions for the skew cavity. Their calculations were both based on structured non-orthogonal grids similar to Figure 5.2. Both authors used a second order discretisation of the convective and diffusive terms in the momentum equations, and calculated their results on a succession of grids with gradually increasing mesh sizes. By varying the size of the grids used in calculation the two sets of published data are both shown to be grid independent, a necessary property of benchmark solutions. The difference between the two calculations was the method used to prevent odd-even de-coupling of the pressure and velocity field. Oosterlee et al.[86] used the staggered mesh approach whereas Demirdzic et al.[85] adopted Rhie and Chow's[20] interpolation method. However, it can be clearly seen from Oosterlee et al.'s[86] data that this difference in the interpolation method has little effect on the results in this test case. In fact the two sets of published data are virtually identical which is a good indication of the suitability of these results for benchmarking purposes.

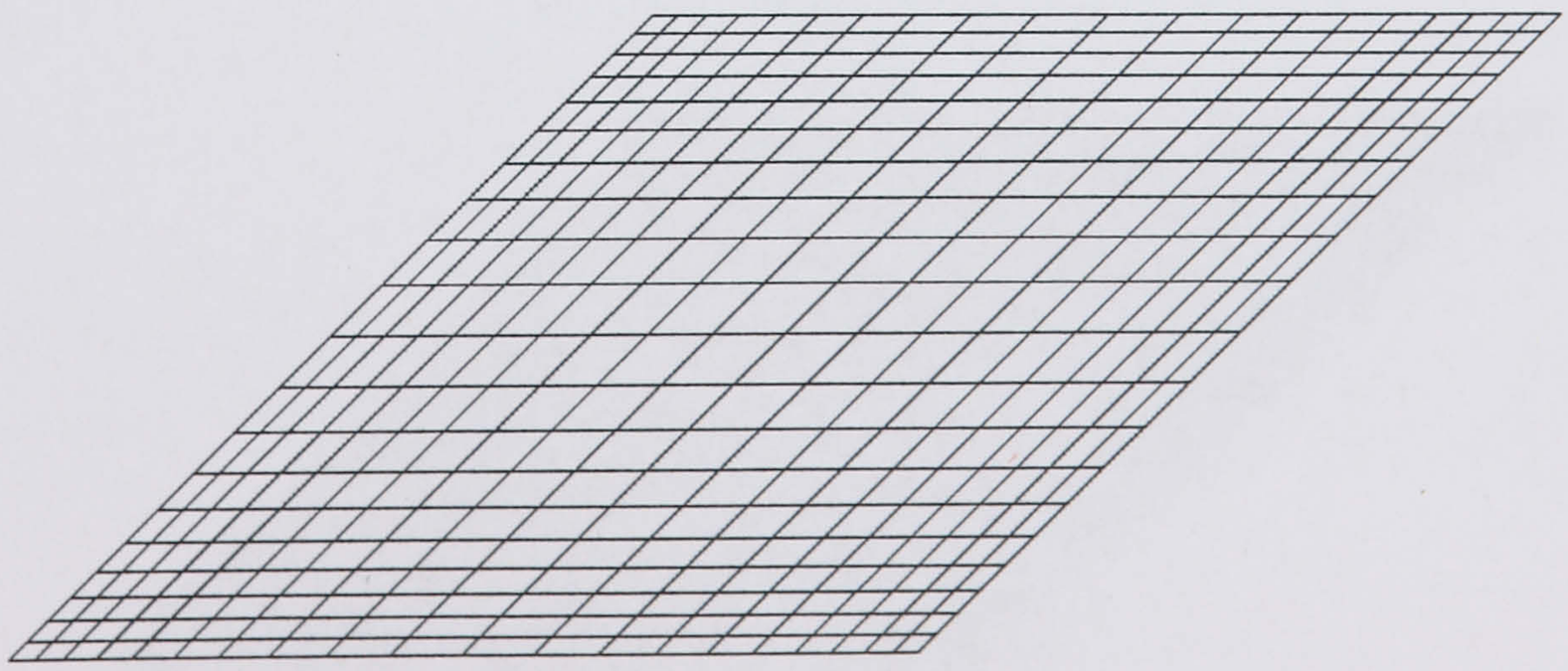
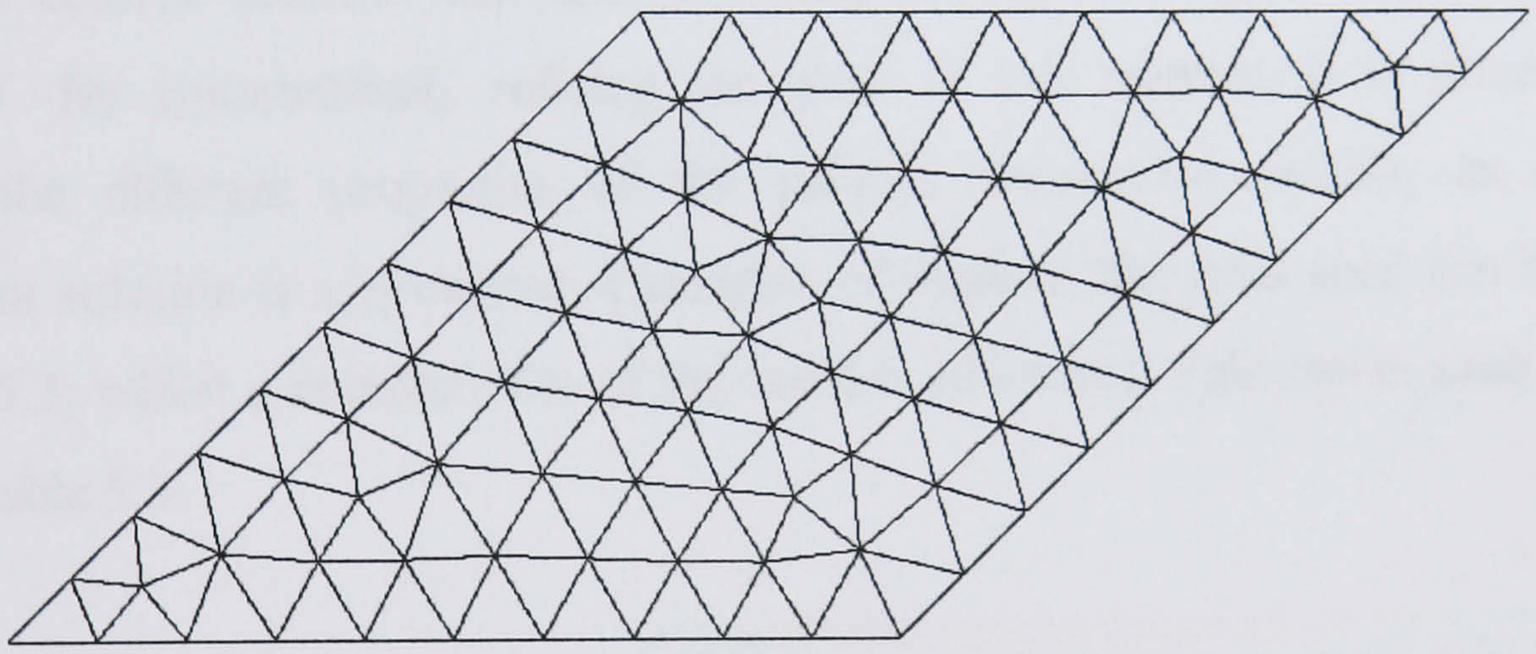
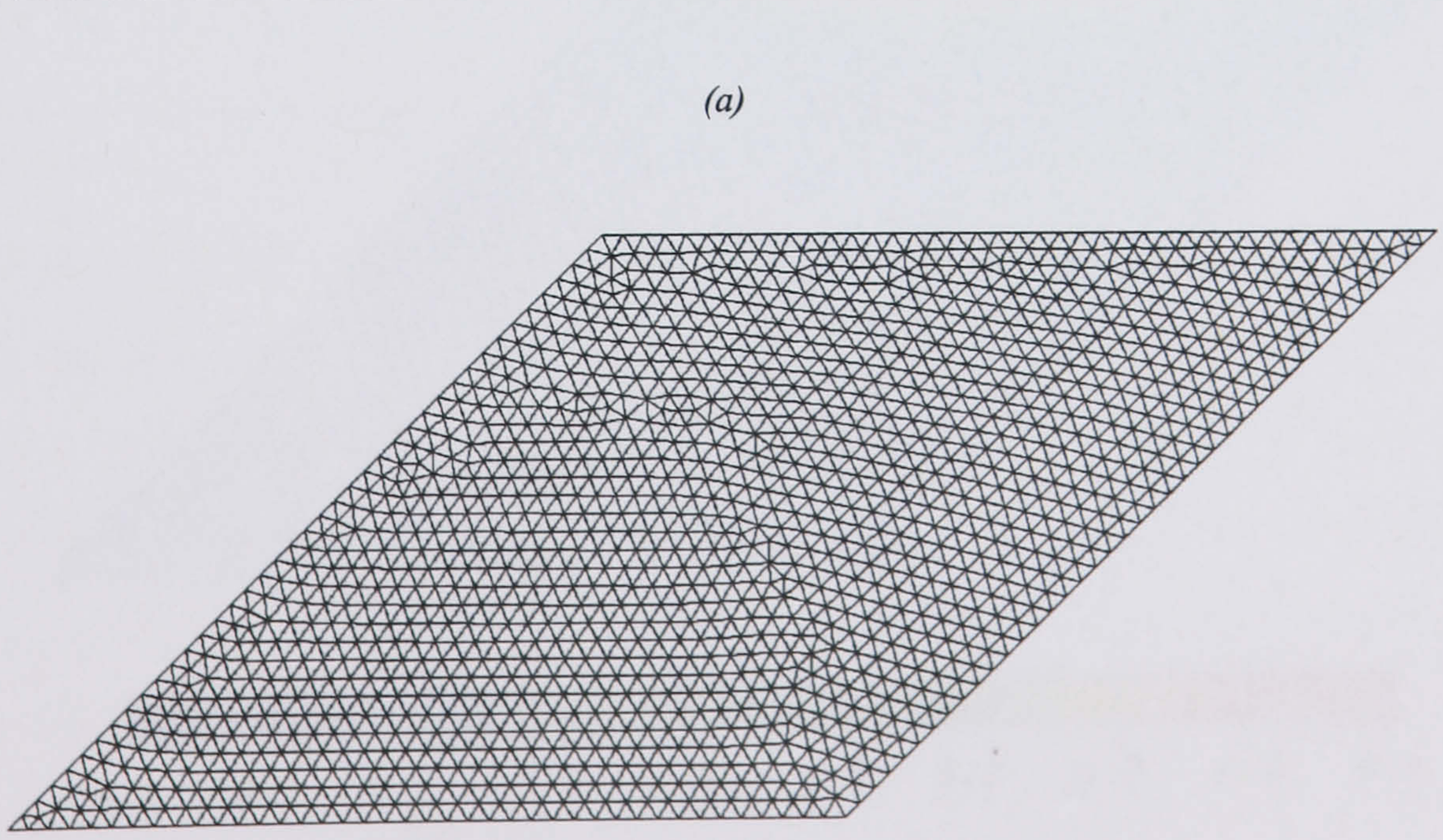


Figure 5.2: Skew Cavity Structured Non-Orthogonal Mesh

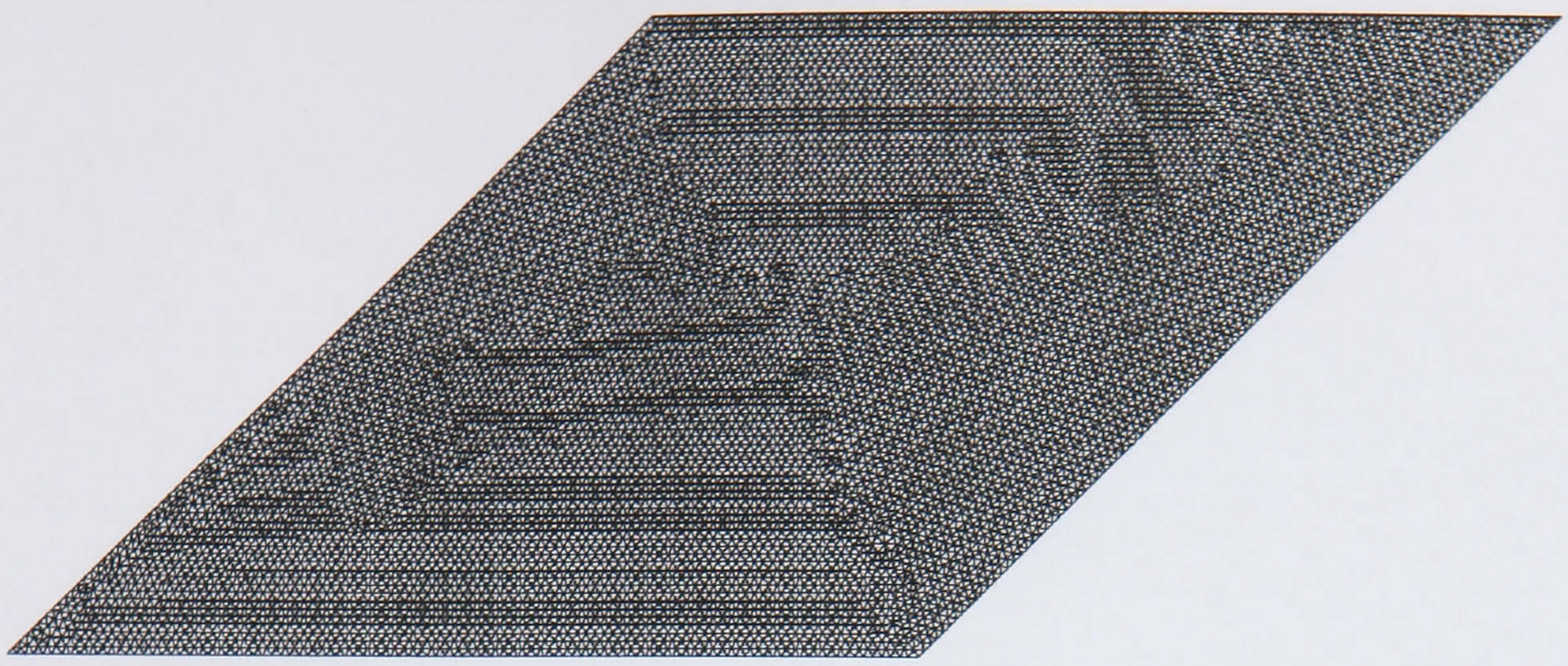
Here, the calculated results are compared to the published data of Demirdzic et al.[85]. As virtually no difference exists between Demirdzic et al.[85] and Oosterlee et al.'s[86] data this choice is made arbitrarily. Both authors calculated two different skew cavity configurations, one with a corner angle of 45 degrees as in Figure 5.1 and the second with the angle decreased to 30 degrees. As the results of both these cases are very similar, only the first of the two cases is calculated here. The Reynolds number of the calculations, defined using the lid velocity, U_L, and the cavity length, L, was set at two separate values, 100 and 1000. For both of these cases L=1, density ρ=1.2 and the lid velocity U_L=1 hence the viscosity, μ, was calculated as 0.012 and 0.0012 respectively. The boundary conditions applied were, $u=U_L$ and v=0 on the lid of the cavity and $u=v$=0 on the three other walls. All the boundary conditions are applications of the no-slip condition and are implemented as described in the wall and inlet boundary condition sections of Chapter 3. The calculations were iterated until the residual of each of the governing equations was reduced to 10^{-4}.



(a)



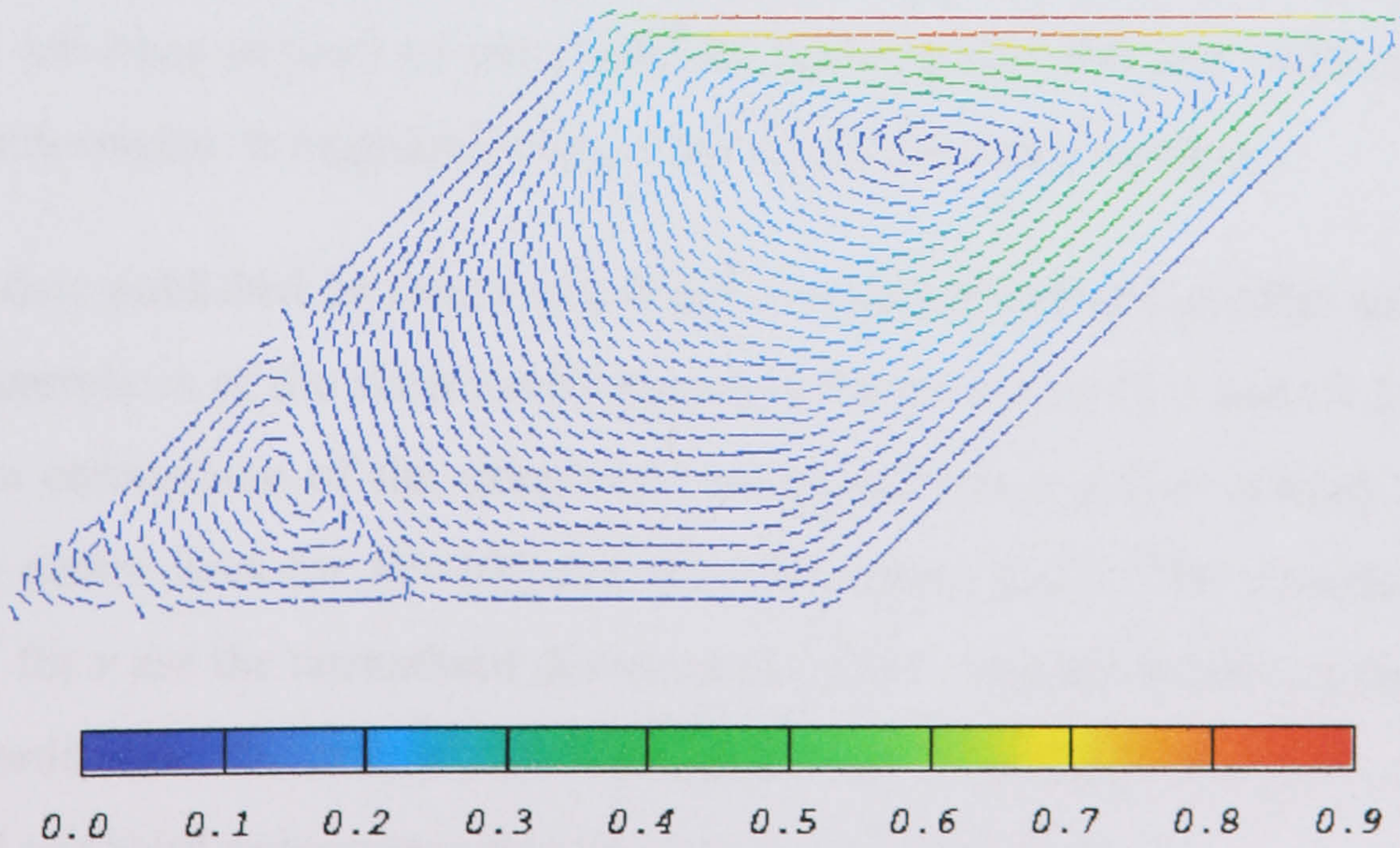
(b)



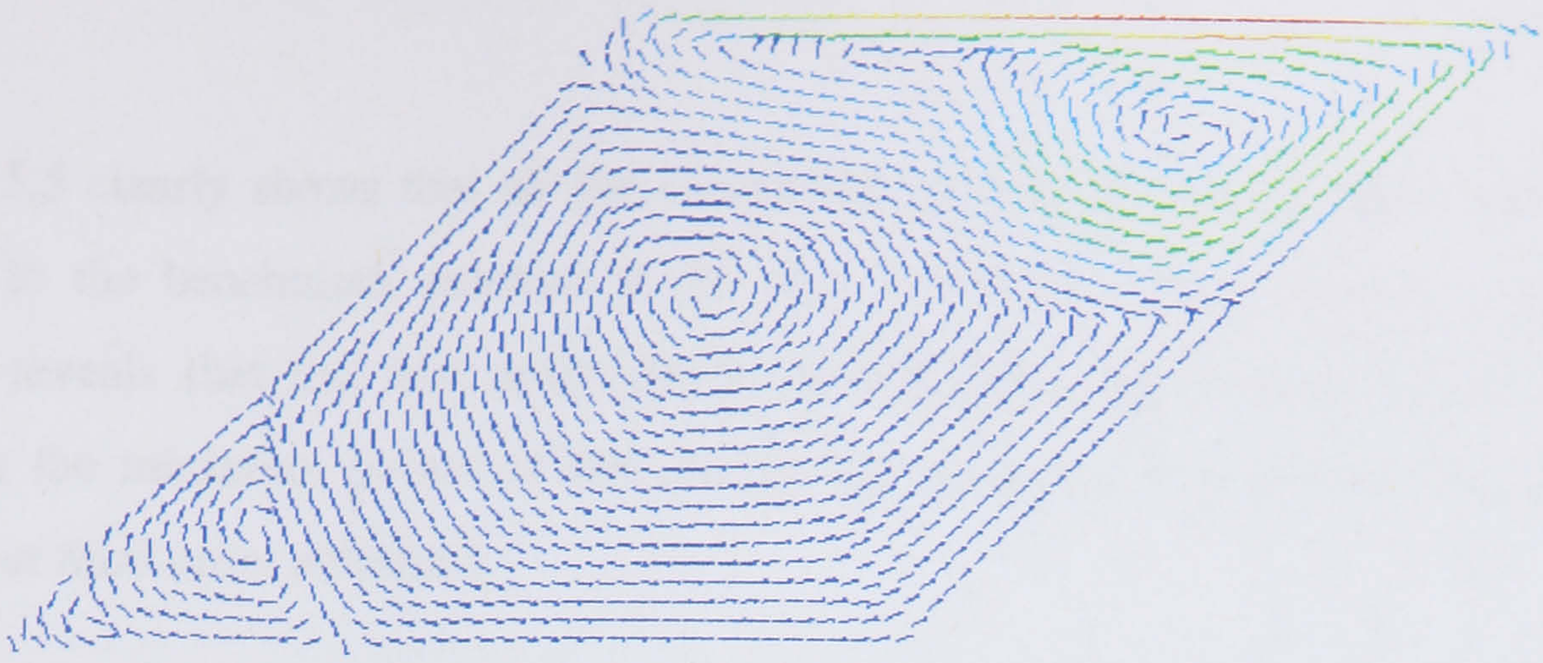
(c)

Figure 5.3: Skew Cavity Triangular Unstructured Meshes (a) 162 CV (b) 2684 CV (c) 16626 CV

Demirdzic et al.[85] calculated results on six different structured grids from 10x10 CV (control volumes) to 320x320 CV. Seven unstructured triangular meshes are used here, in order to compare the accuracy of results calculated using unstructured meshes to their structured counterparts. The smallest grid contains 162 CV and the number of control volumes increases gradually to the finest grid which contains 16626 CV. By systematically refining the grids in this manner, it is possible to compare the different properties of the various convective models, as a grid independent solution is approached. Examples of three of the grids used can be seen in Figure 5.3, whilst a complete list of the number of control volumes in each grid is given in Table 5.3.



(a)



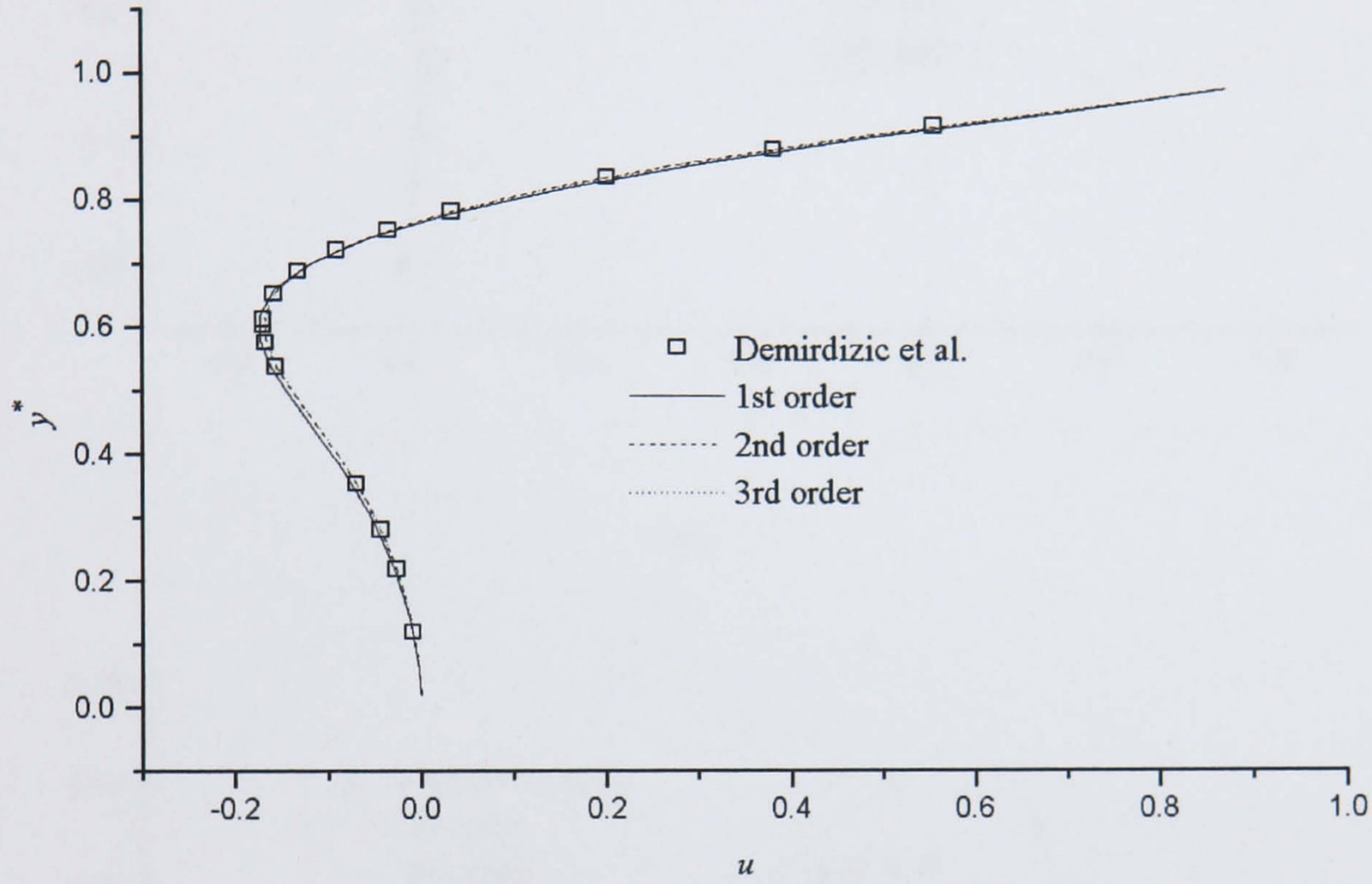
(b)

Figure 5.4: Velocity Vectors in the Skew Cavity Coloured by their Magnitude (m/s) (a) Re=100 (b) Re=1000

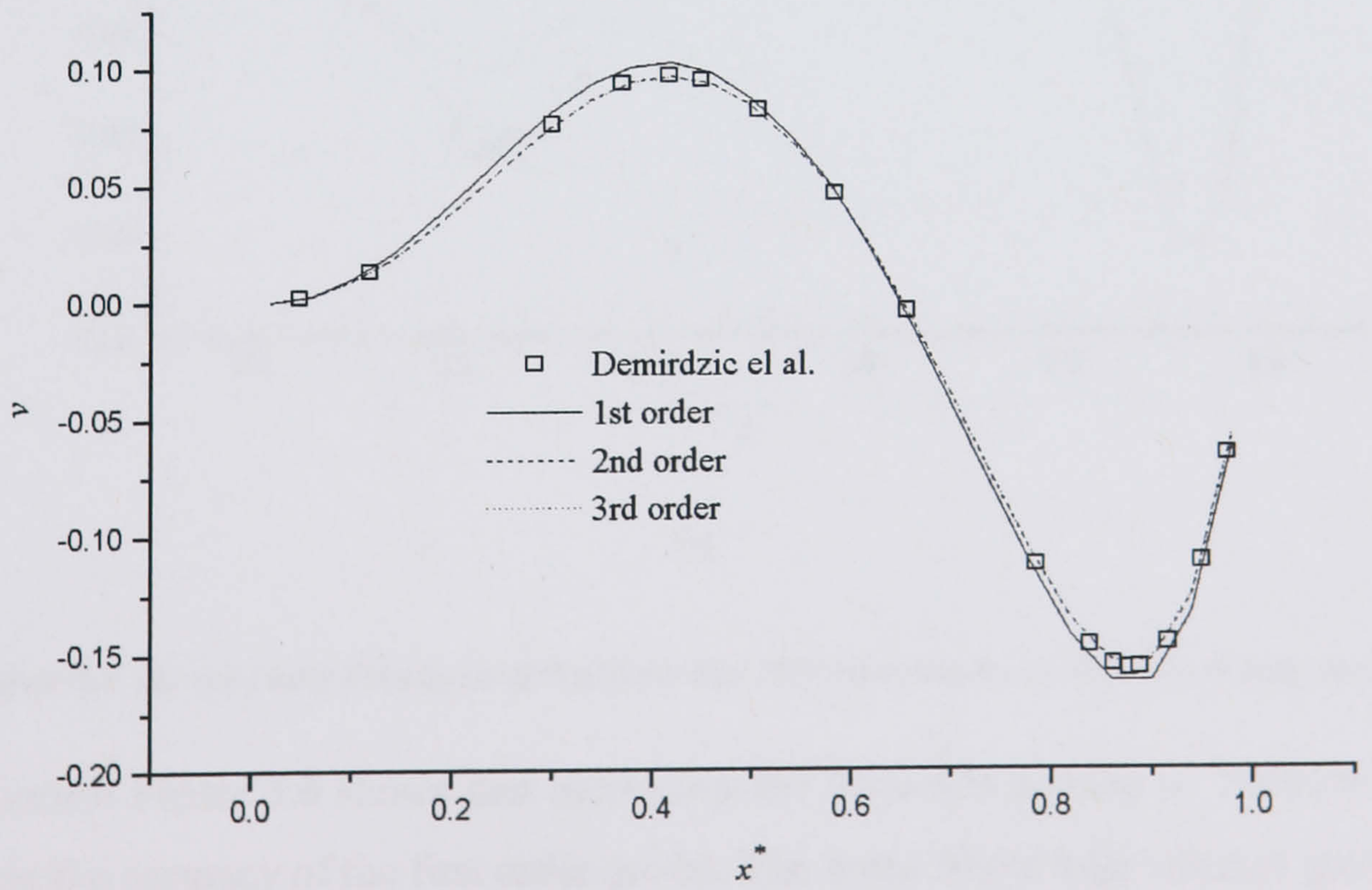
Figure 5.4(a) and Figure 5.4(b) shows the velocity vectors calculated on the 2684 CV grid using second order convective discretisation for a Reynolds numbers of 100 and 1000 respectively. For Reynolds number 100 the main vortex fills almost the whole of the cavity. A second counter-rotating vortex lies deep in the bottom left-hand corner of the cavity but it is much smaller in strength. When the Reynolds number is increased to 1000 the lower viscosity causes a very different flow pattern. The primary vortex now only fills approximately one third of the cavity in its top right-hand corner whilst a second much larger counter-rotating vortex, in comparison to the *Re*=100 case, fills the central area of the cavity. Smaller vortices are predicted in the bottom left-hand corners of the cavity for both the *Re*=100 and 1000 cases but they are much smaller in magnitude than either of the two main vortices.

Part of the data published by Demirdzic et al.[85] consists of velocity profiles calculated along the centrelines of the skew cavity shown in Figure 5.1 as CL1 and CL2. Figure 5.5 shows a comparison of the calculated and benchmark u and v velocity profiles along these centre lines for *Re*=100 and *Re*=1000 respectively. The co-ordinates y^* for u and x^* for v are the normalised distances along the centrelines and not the actual x and y co-ordinates. The calculations were performed on the 2684 CV grid using the first, second and third order convective models detailed in Chapter 3.

Figure 5.5 clearly shows that all the convective models produce virtually identical results to the benchmark solution at *Re*=100. Close inspection of the *v* velocity profile reveals that the first order model over predicts the maximum and under predicts the minimum points of the profile but otherwise it is as accurate as the second or third order schemes.

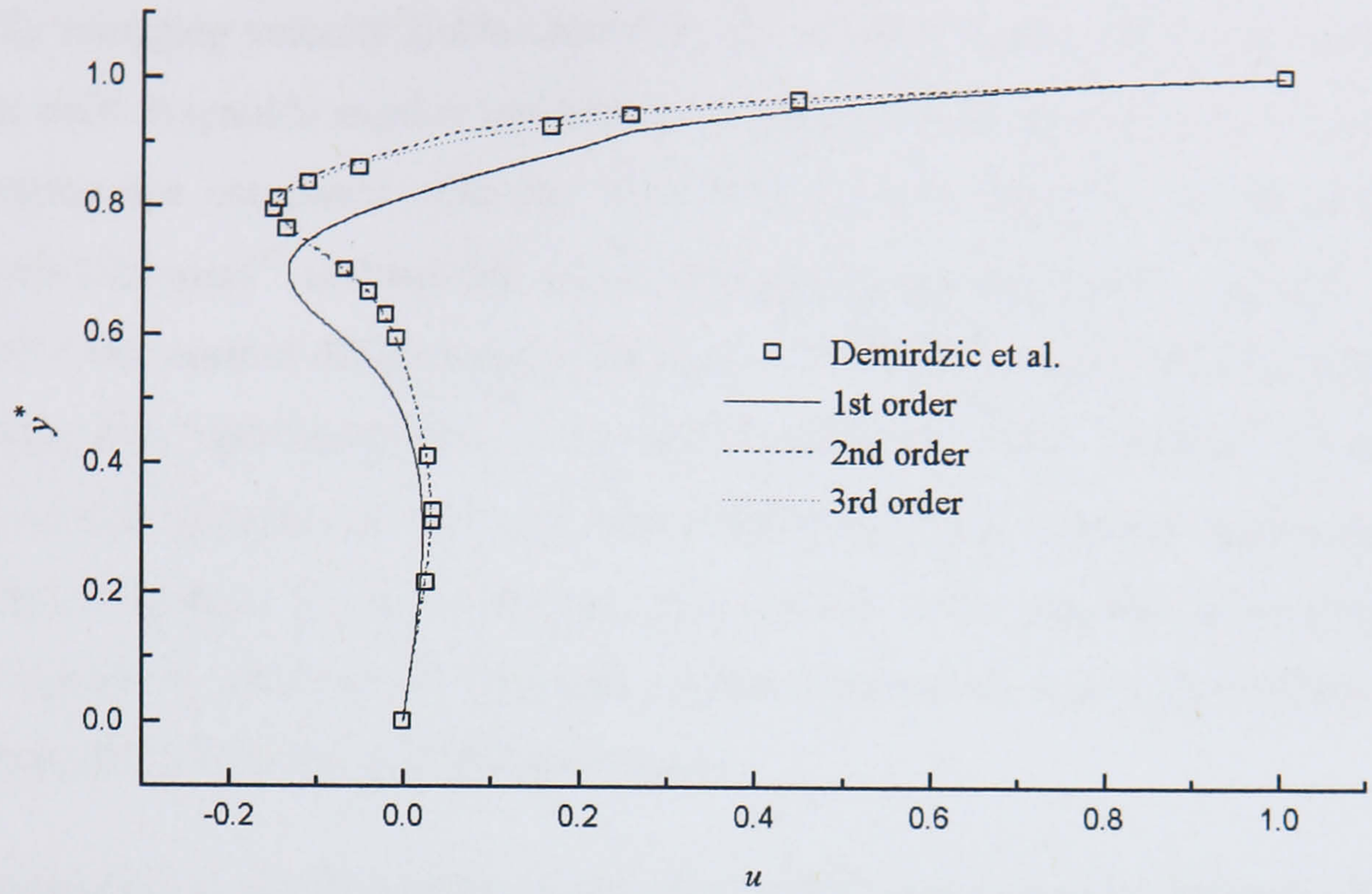


(a)

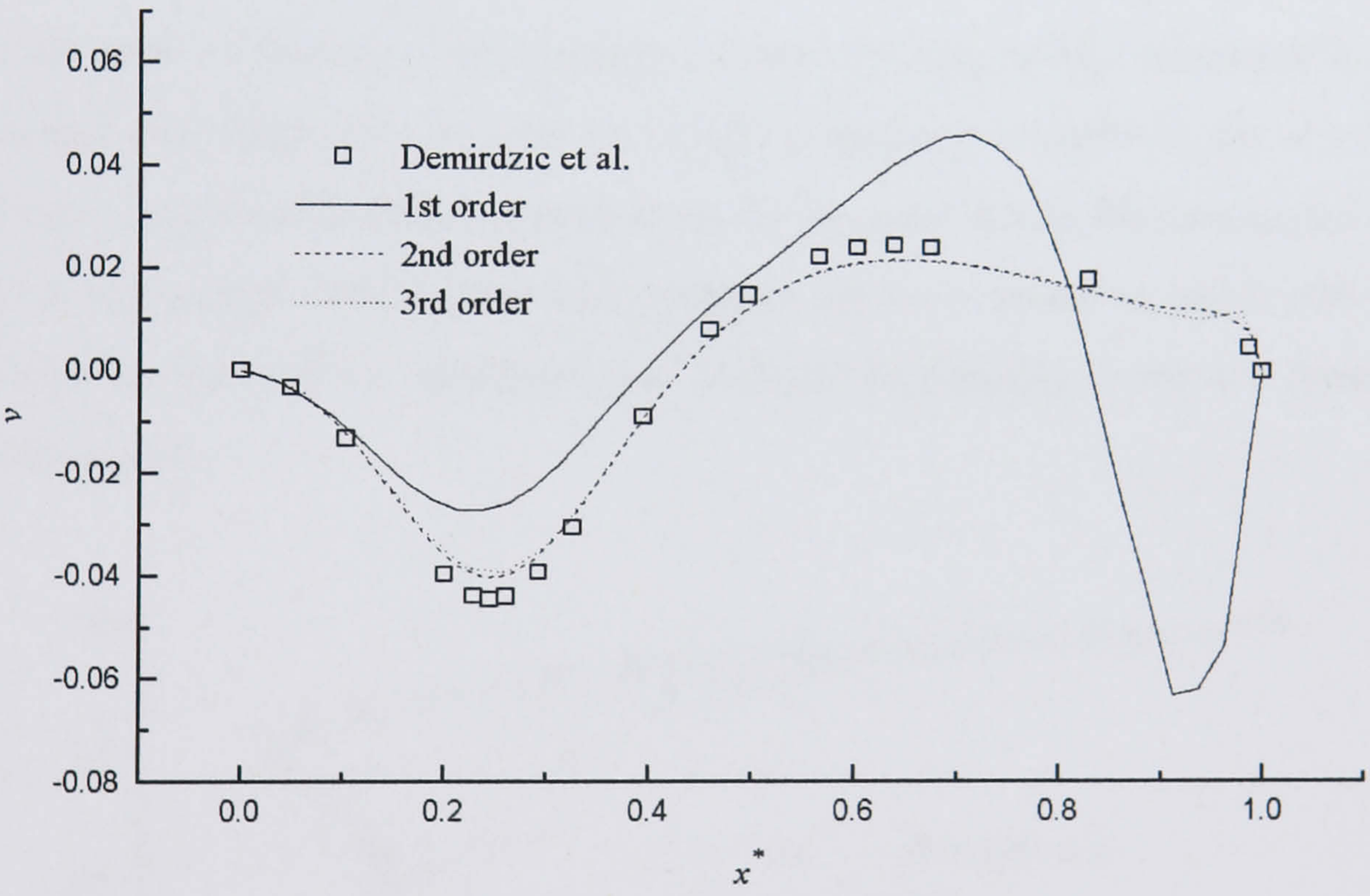


(b)

Figure 5.5: Skewed Cavity Velocity (m/s) Profiles at Re=100 (a) x-velocity on CL1 (b) y-velocity on CL2



(a)



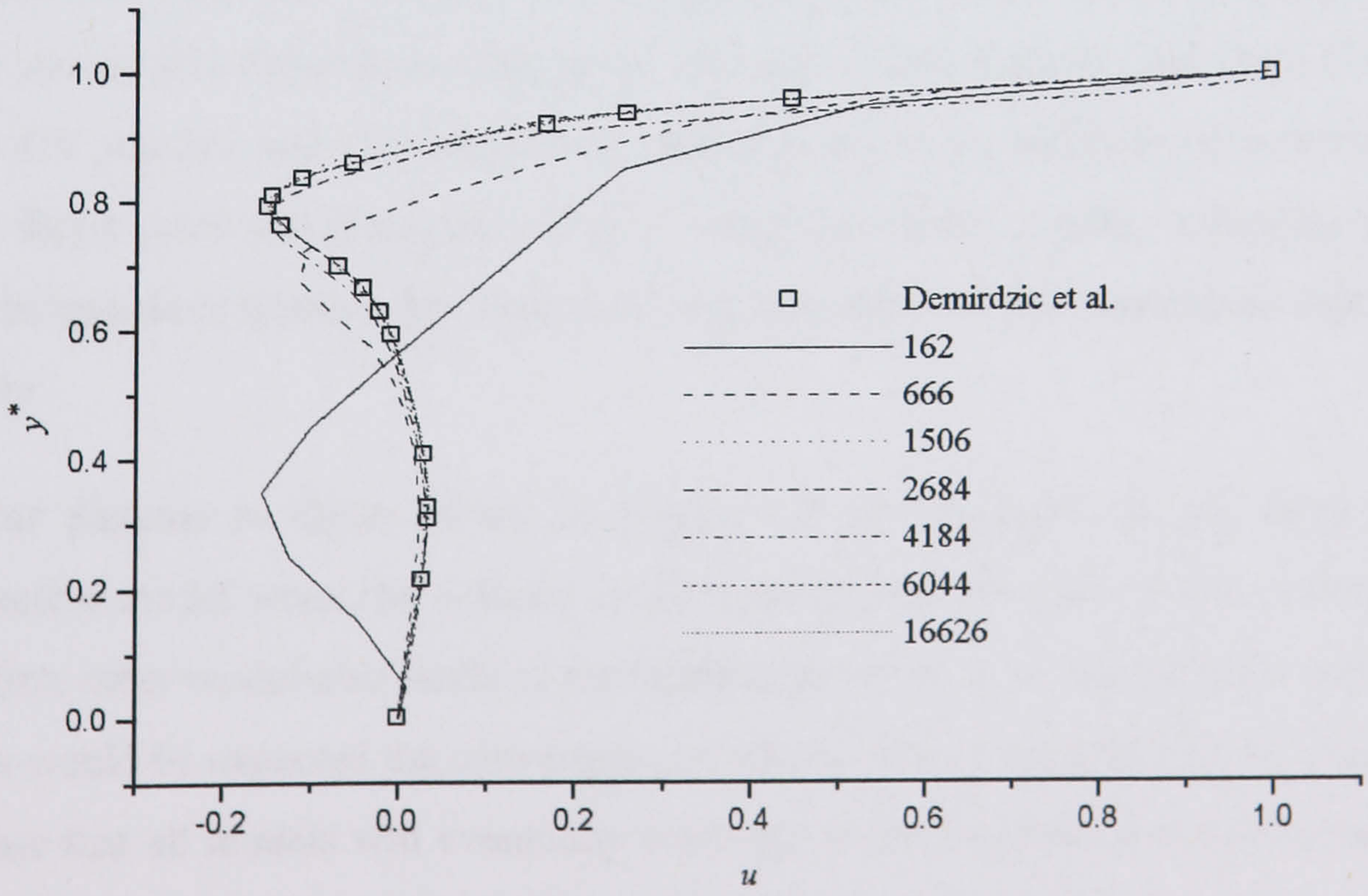
(b)

Figure 5.6: Skewed Cavity Velocity (m/s) Profiles at Re=1000 (a) x-velocity on CL1 (b) y-velocity on CL2

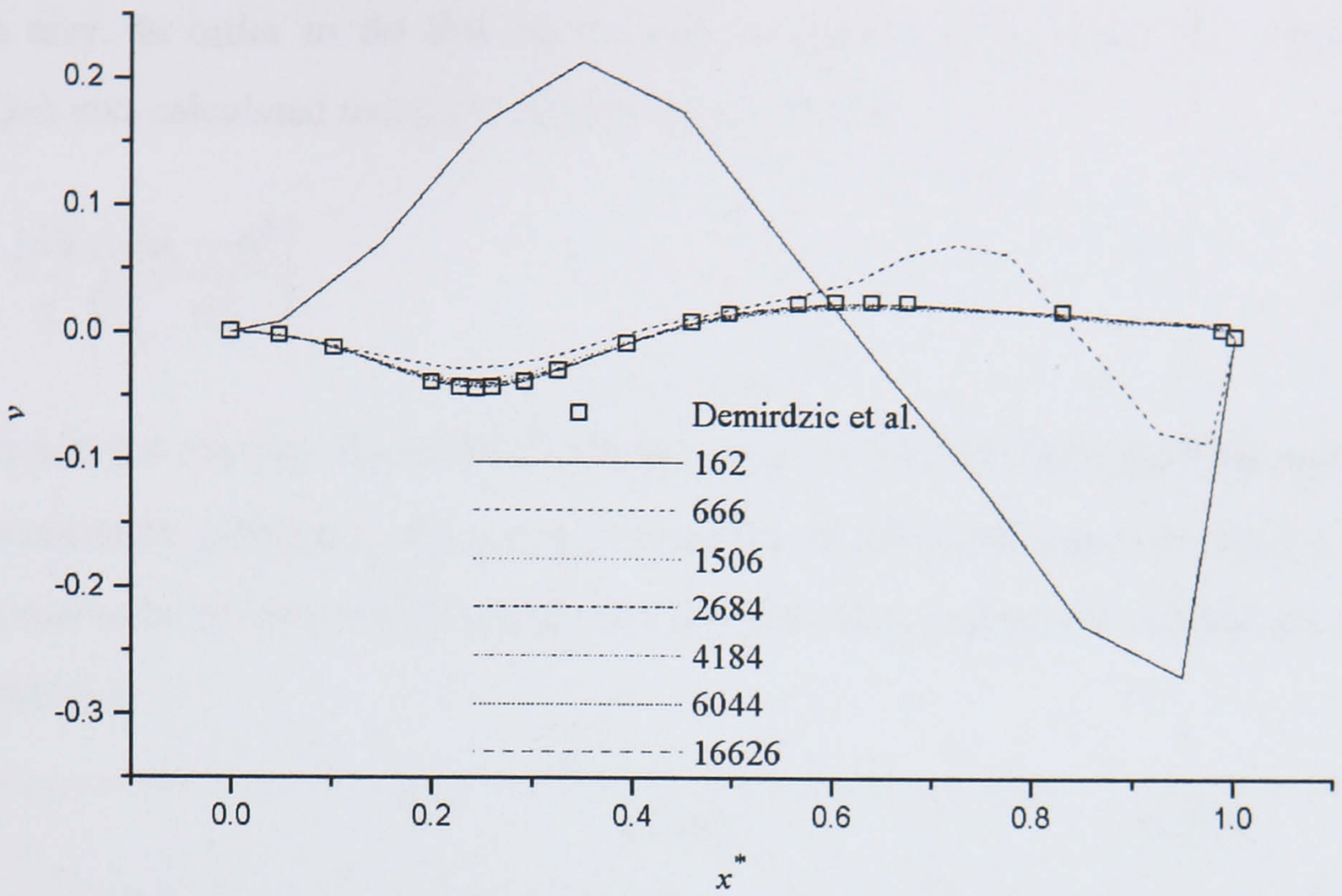
In contrast Figure 5.6 shows that increasing the Reynolds number to 1000 severely effects the accuracy of the first order model. The areas where high velocity gradients exist are those that are most effected by the deficiencies of first order modelling. Hence in both velocity profiles it is the areas where the centrelines pass through the

rapidly changing velocity fields caused by the vortices in the cavity that the largest errors exist. Reynolds number and high gradients effecting the accuracy of first order modelling are consistent with the idea of numerical diffusion. As described in Chapter 2 Leonard[17] showed that a first order upwind model could be thought of as a second order centred differencing scheme with an additional gradient term added for stability. This stabilising term or numerical diffusion would obviously become a large in the presence of high gradients but would also increase with Reynolds number. Therefore it is not surprising that the first order upwind model produces such inaccurate results as the Reynolds number is increased, and in the regions of the solution where high velocity gradients exist.

In comparison to the first order model, the second and third order upwind schemes both produce results that match the benchmark solution closely at Re=1000. Figure 5.6 shows that the results of two high order models do not differ greatly despite their differing order of accuracy. The extrema of the v velocity profile calculated by both the second and third order models are slightly inaccurate whilst in the u velocity profile the second order model over predicts the increase in u as the moving lid of the cavity is approached. Whilst these discrepancies are by no means as inaccurate as the first order model they are significant and it should be possible to remove them with grid refinement.



(a)



(b)

Figure 5.7: Variation of the Centreline Velocities (m/s) as a Function of Grid Fineness for Re=1000 (a) x-velocity on CL1 (b) y-velocity on CL2

To test the hypothesis that a more accurate solution will be possible with grid refinement, Figure 5.7 shows a comparison of the u and v velocity profiles along the centrelines from all the seven grids used at Re=1000. These profiles are calculated using the second order convective model, and clearly show that as mesh size increases the solutions converge to the benchmark solution. The two coarsest grids show strong grid dependence and errors are also visible between the 1506 CV and 2684 CV meshes and the benchmark solution. However, differences between the three finest grids and Demirdzic et al.'s[85] data are hardly visible, indicating that a grid independent solution has been achieved that matches the benchmark data very closely.

Similar patterns to those shown in Figure 5.7 are produced by the third order convective model when the velocity profiles are plotted for each of the seven grids. The first order model also tends to the benchmark solution as the grid size increases but as would be expected the convergence is slower. However, although these graphs indicate that all models will eventually converge to the benchmark solution, they do not give a quantitative way of evaluating which model is most accurate for a given

mesh size. In order to do this an average percentage error from the benchmark solution was calculated using the following expression

$$\varepsilon_{\phi} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\phi_i - \phi_i^B}{\phi_i^B} \right| \qquad (5.1)$$

where n is the number of published data points, ϕ_i is the calculated solution and ϕ_i^B is the benchmark solution at the given data points. In this test case ϕ is either u or v. The results for the seven different mesh sizes and three convective models are given in Table 5.3.

Grid Size	ε_u (%)			ε_v (%)		
	1st order	2nd order	3rd order	1st order	2nd order	3rd order
162	540.6	321.9	306.6	870.2	789.3	757.3
666	172.8	127.9	121.1	209.0	135.3	122.9
1506	141.9	16.2	15.9	79.6	20.5	19.1
2684	143.1	12.0	10.5	70.7	13.4	12.1
4184	133.8	4.6	4.4	60.0	8.7	7.5
6044	114.1	2.7	2.6	44.2	6.1	5.9

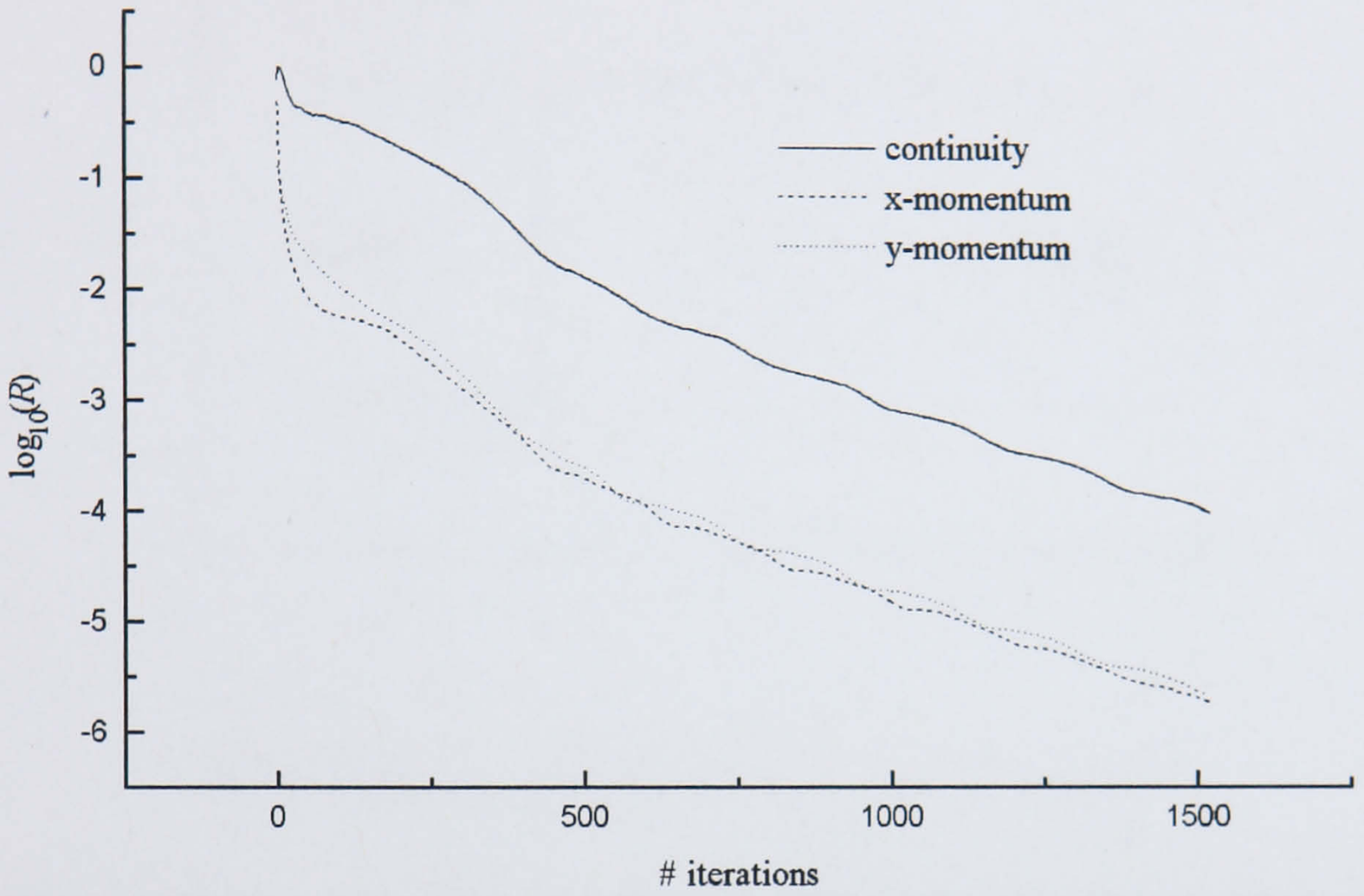
16626	69.3	3.6	3.5	23.8	3.51	3.02

Table 5.3: Average Percentage Errors for the Skew Cavity Test Case at Re=1000

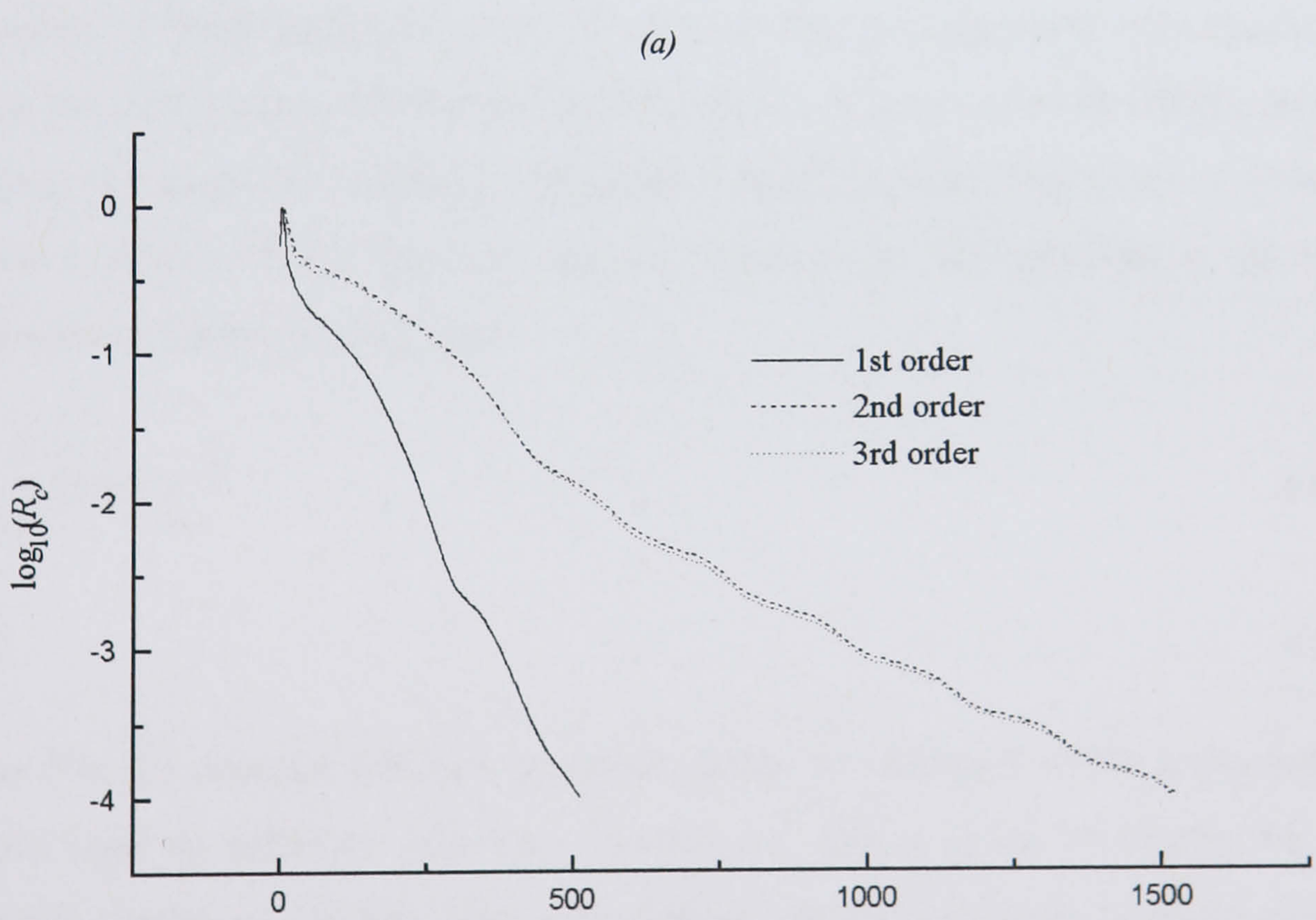
This table clearly shows the relative accuracy of the various convective models. The first order model produces the least accurate solutions followed by the second order scheme and the most accurate results are produced by the third order model. As the grid size increases the difference between the second and third order model's errors quickly reduces whilst the error in the first order scheme remains significantly higher than the other two models. These properties of the three models are to be expected and can be understood by considering the magnitude of the terms discarded when the Taylor series was truncated to form the different schemes. The errors in the first order model should reduce linearly, whilst the second and third order scheme's errors should decrease in a quadratic and cubic nature respectively. As the convective model only forms one part of the discretisation of the Navier-Stokes equations this is not rigidly the case here but the general pattern is certainly in evidence.

Figure 5.8(a) shows the convergence history of all three governing equations for the *Re*=1000 case using the 2684 CV grid and a second order convective model. Both momentum equations converge at virtually the same rate whilst the continuity equation takes approximately three times the number iterations of the momentum equations. Figure 5.8(b) shows a comparison of the convergence history of the continuity equation for the three convective models. Again the convergence data is taken from the *Re*=1000 case using the 2684 CV grid. The continuity equation was chosen to compare convergence histories because it was the equation that took the most iterations to converge. Both the second and third order models converge in approximately three times the number iterations that the first order model takes. This can be accounted for by two factors. Firstly, by using the higher order models the momentum equations become more explicit in nature, and therefore more iterations can be expected to achieve a convergent solution. Also the stabilising effect of numerical diffusion allows higher relaxation factors to be used when iterating to a solution using the first order model. In this case α=0.2 and E=4.0 was used for the first order model and α=0.1 and E=1.0 for the second and third order schemes. It is

interesting to note that the third order model does not require a significantly greater number of iterations to converge than the second order model despite the more explicit nature of the equations.



(a)



(b)

Figure 5.8: Skew Cavity Convergence Histories (a) Comparison between Governing Equations (b) Comparison between Convective Models

5.3 Poiseuille Flow between Two Flat Plates

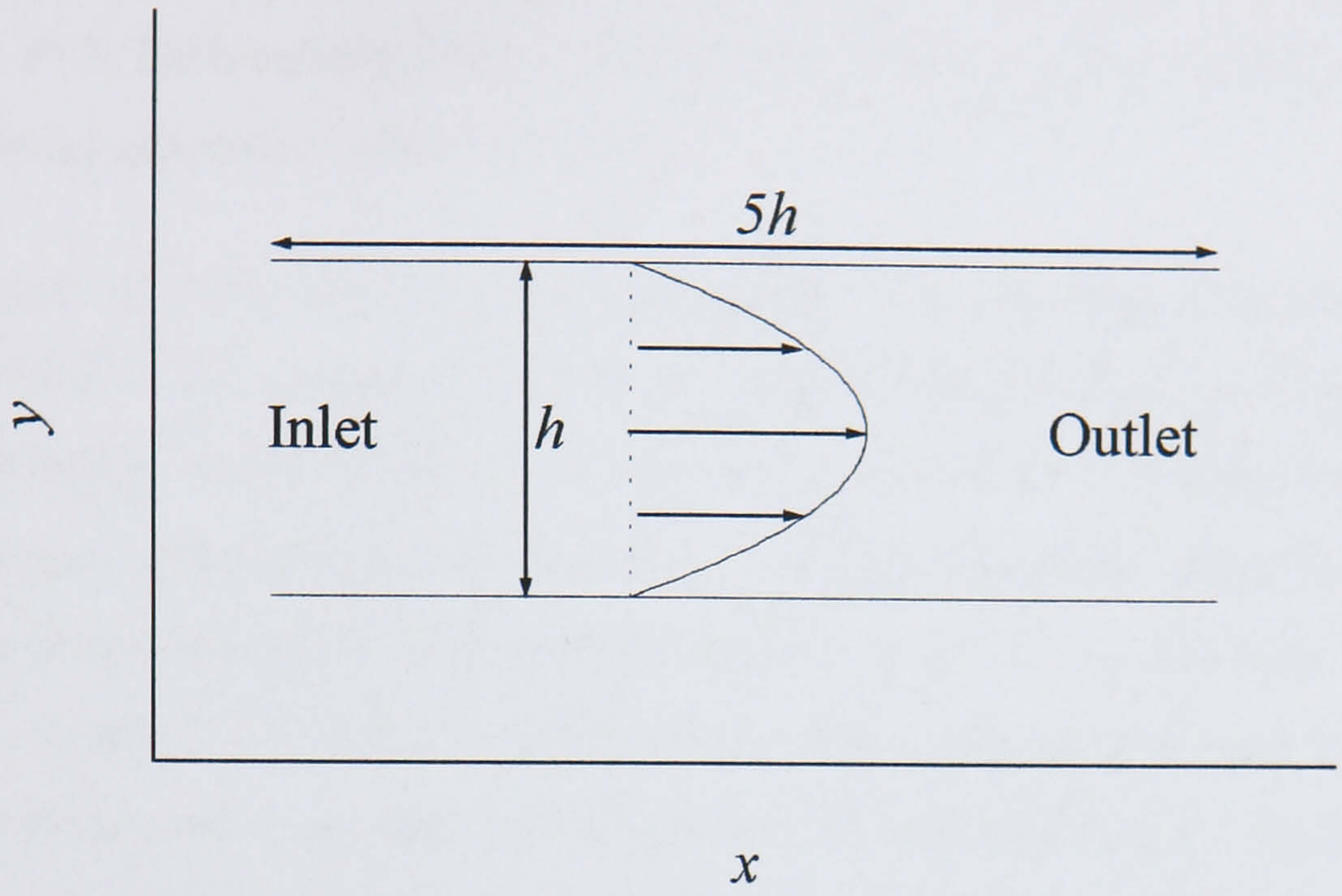


Figure 5.9: Schematic of the Poiseuille Flow between Two Flat Plates

Poiseuille flows are generated by pressure gradients and named after J.L.M. Poiseuille a French physician who first studied them in connection with blood flow. In this test case a constant pressure gradient, dp/dx, is imposed on the fluid contained between two parallel flat plates a distance h apart perpendicular to the y direction. Classical theory shows that an analytic solution to this problem is given by expressions of the following form

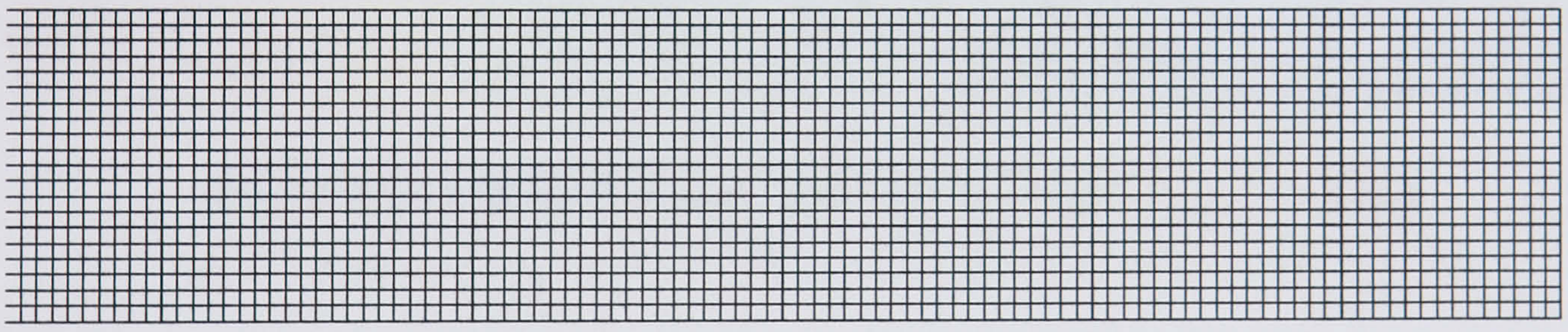
$$u = \frac{P}{2\mu}\left(yh - y^2\right) \qquad (5.2)$$

$$v = 0 \qquad (5.3)$$

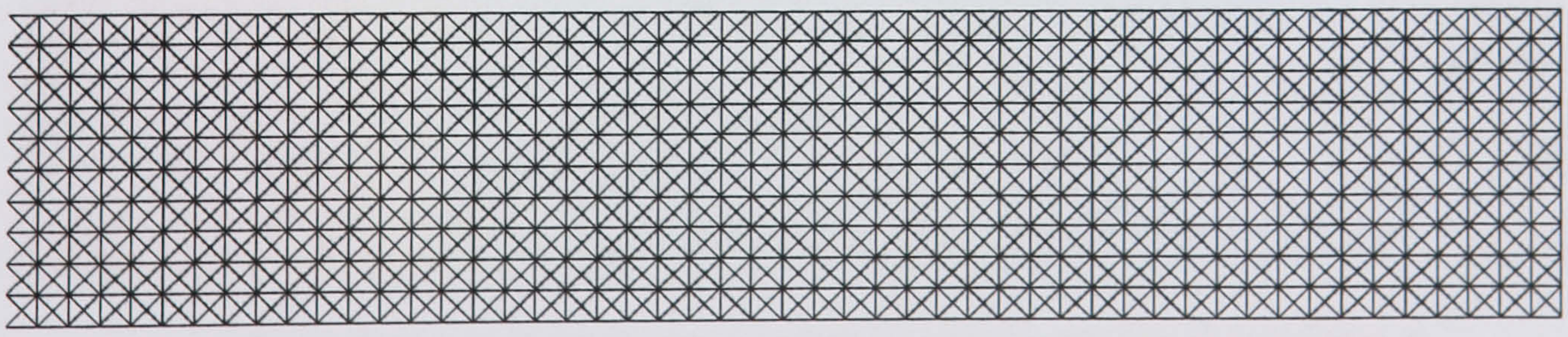
where P is the constant pressure gradient $-dp/dx$. A schematic of the computational domain used to calculate solutions to this test case is given in Figure 5.9. The boundary conditions applied were, u=v=0 along the parallel plates, u and v given by equations (5.2) and (5.3) respectively at the inlet and either a constant pressure or derivative boundary condition at the outlet of the domain. In the cases where a derivative boundary condition is applied the pressure is set to zero at the centre of the computational domain. When a constant pressure at the outlet of the domain is prescribed then this value is also taken to be zero. The Reynolds number of the

calculations was set at 100 based upon the distance between the parallel plates, h=1, and the average inlet velocity, U_{avg}=1. The density, ρ=1.2, and the constant pressure gradient, P=1. Each calculation performed was iterated until the residual of each of the governing equations was reduced to 10^{-4}.

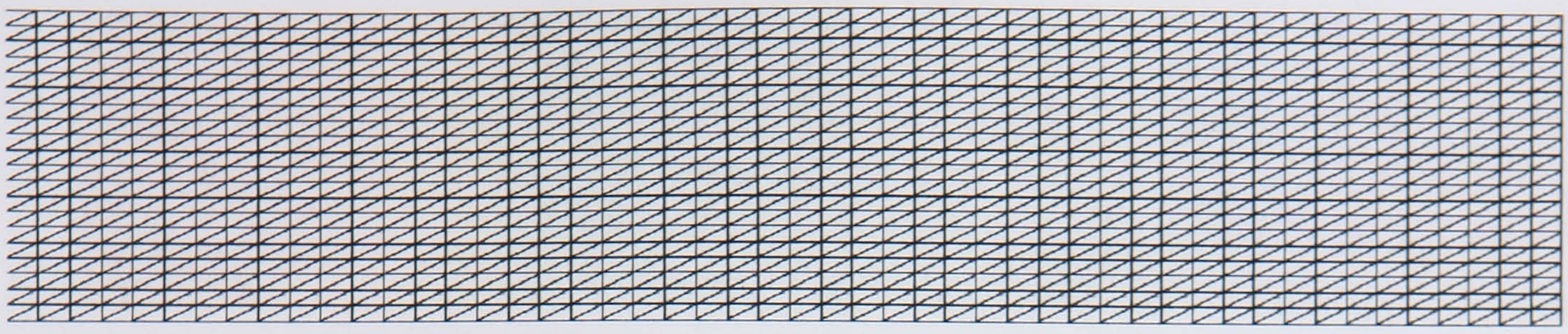
The purpose of using this test case is to analyse the calculated pressure field and highlight some of the inaccuracies that can be caused to this pressure field if certain outlet boundary conditions are used. Also a comparison will be made between the time taken and computational overheads of solutions calculated using Gauss-Seidel point by point iteration or algebraic multigrid to solve the pressure correction equation. Figure 5.10 shows examples of the four types of grid used to calculate solutions to this test case. The first is a structured orthogonal mesh, the second and third consist of regular isosceles and right-angled triangles respectively, and the fourth is a completely unstructured triangular mesh. Not only do these three different forms of mesh show the flexibility of the solution algorithm to adapt to a wide range of mesh types but they will also highlight how the derivative boundary condition is not suitable as an outlet boundary condition on unstructured grids.



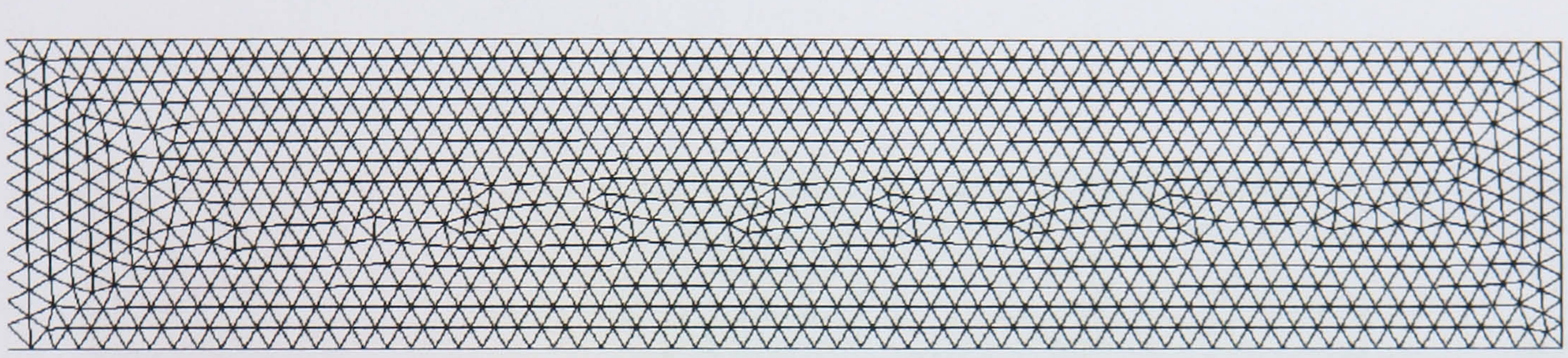
(a)



(b)



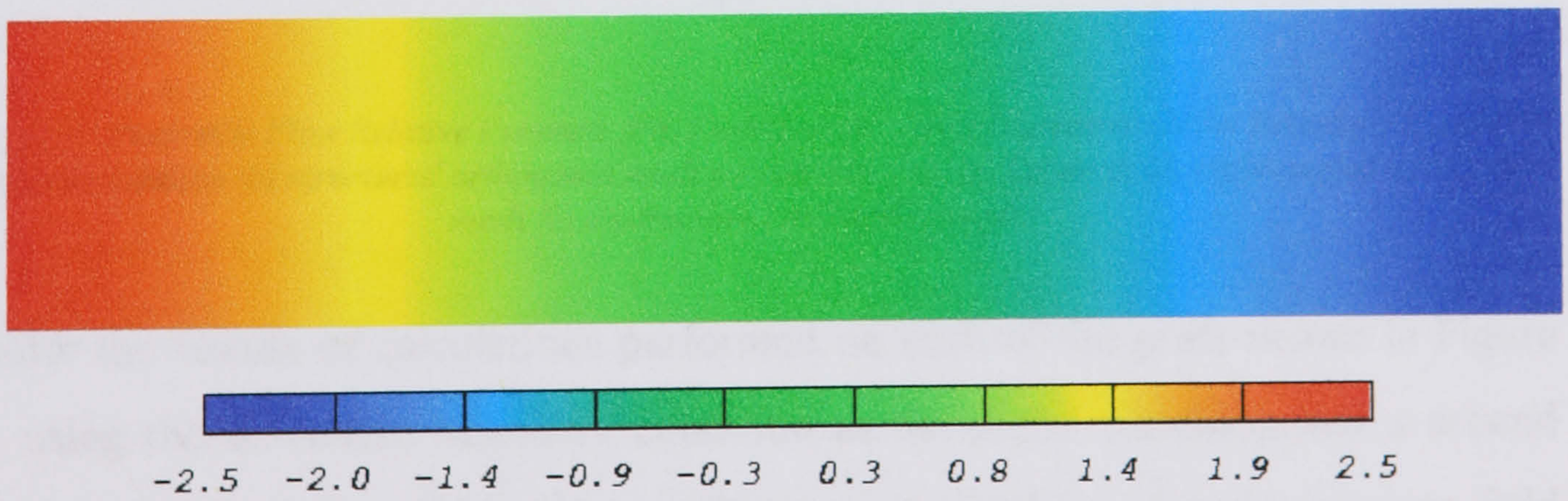
(c)



(d)

Figure 5.10: Poiseuille Flow Grids (a) structured orthogonal (b) isosceles triangular (c) right-angled triangular (d) unstructured triangular

(a)



(b)

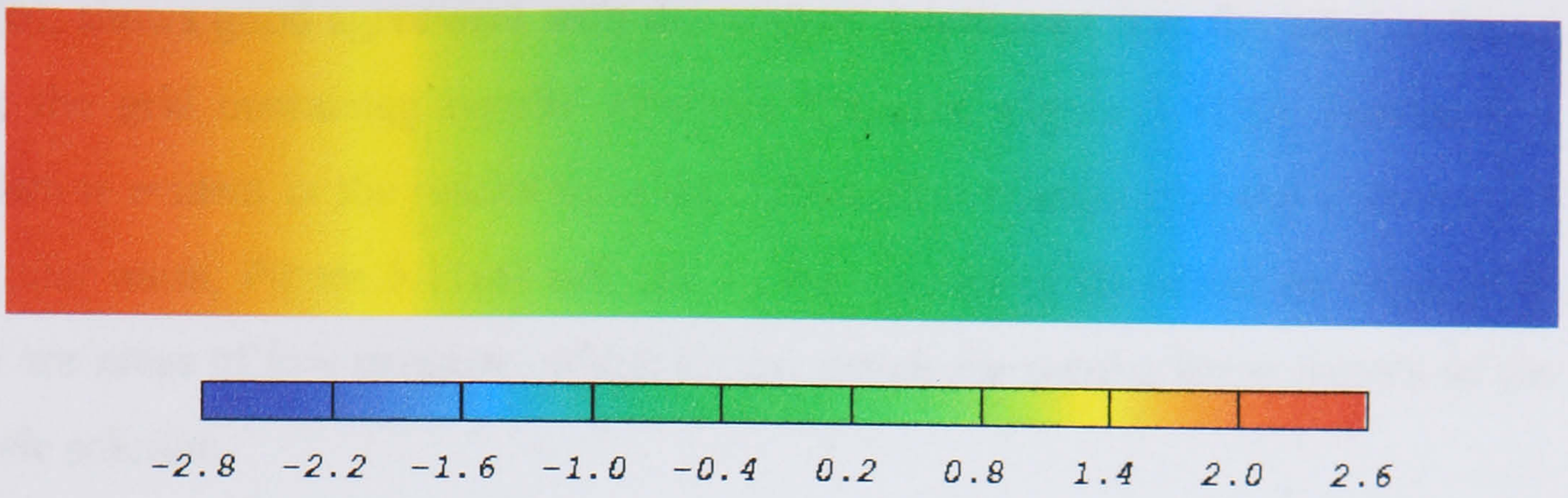


Figure 5.11: Poiseuille Flow Relative Pressure (Pa) Distributions using Derivative Outlet Boundary Condition (a) analytic solution (b) structured orthogonal mesh (c) isosceles triangular mesh (d) right-angled triangular mesh (e) unstructured triangular mesh

Consider the results of calculations performed on each of the grids shown in Figure 5.10, using the derivative boundary condition as the outlet condition and a second order convective model. From the conditions prescribed at inlet the pressure field should be a linear distribution in the x direction dropping in value by 5 from inlet to outlet. Figure 5.11 shows the calculated pressure distributions from the four grids and the analytic solution. The results from the structured orthogonal grid, Figure

5.11(b), shows good agreement with the analytic solution as does the solution found using the grid containing regular isosceles triangles, Figure 5.11(c). However, a problem is evident in the results from the right-angled triangle grid and unstructured triangular mesh, Figure 5.11(d) and (e). In the corners of the outlet on these grids there are areas of low pressure, which do not match the regular linear pattern of the analytic solution.

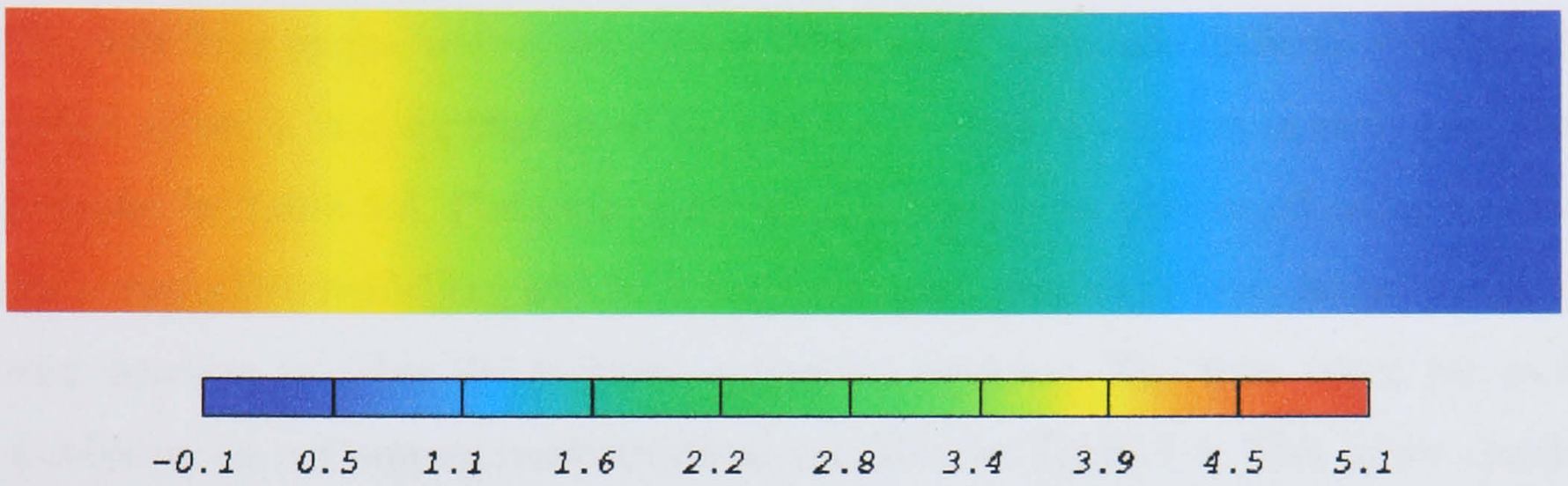


Figure 5.12: Poiseuille Flow Relative Pressure (Pa) Distribution on an Unstructured Triangular Mesh using Fixed Pressure Outlet Boundary Condition

Further investigation into this problem revealed that the interpolation of the pressure field from the cell nodes to the cell faces was the cause of the problem. The interpolation method described in Chapter 3 was simply that the face pressure should be set equal to value of pressure at the adjacent node. Clearly this method is perfectly adequate for grids where the vector joining the boundary face node to the adjacent cell node is perpendicular to the boundary, as is the case with the structured orthogonal mesh and the grid containing regular isosceles triangles. However where this is not the case as in the remaining two example meshes the interpolation scheme is clearly highly inaccurate. Implementing a more accurate interpolation scheme would appear to be an obvious solution but a number of different schemes were tested and none of them removed the inaccuracies entirely. Given that interpolating seemed to be a fundamentally flawed method of finding the pressure on the outlet boundary a change in boundary condition seemed appropriate. Therefore, instead of using the derivative boundary condition at the outlet the fixed pressure condition, described in Chapter 3, was applied. The resulting pressure distribution calculated on the unstructured triangular mesh can be seen in Figure 5.12. Clearly the inaccuracies at the corners of the outlet are completely removed and the pressure field is very close to the analytic solution. No disadvantages are entailed from using the fixed

pressure boundary condition instead of the derivative condition and consequently adopting it as the standard outlet condition is sensible.

The effect of using algebraic multigrid to solve the pressure correction equation on the time taken to arrive at a converged solution was investigated on four separate meshes. One of the grids was the unstructured triangular mesh shown in Figure 5.10. The other three grids were the same form of mesh but contained different numbers of control volumes. A complete list of the number of control volumes in each grid used is shown in Table 5.4. For each mesh a calculation was performed using second order convective modelling and both algebraic multigrid and Gauss-Seidel point by point iteration to solve the pressure correction equation. The time taken for each calculation on a Compaq workstation is recorded in Table 5.4. This table clearly shows what a significant difference to the convergence times the algebraic multigrid makes. The effect of the multigrid is more significant as the mesh size increases because the convergence times of the pressure correction equation will only go up linearly with mesh size whilst the Gauss-Seidel method will increase at a much higher rate. Although the times displayed are not prohibitively large in this simple two-dimensional case for three-dimensional calculations, where typical grids can contain 150,000 control volumes or more, it is obvious that the use of multigrid will be invaluable. There is however a price to pay for the use of algebraic multigrid, increased memory requirements. Clearly, the extra topology information for the multigrid method has to be stored in additional arrays which are not required by the simple Gauss-Seidel method. Therefore the memory requirements of the multigrid scheme are considerably larger than those of the Gauss-Seidel method. Nevertheless, this increase in memory is well within the limits of modern computing resources and therefore can be regarded as acceptable.

Grid Size (CV)	Time to Convergence (s)	
	Multigrid	Gauss-Seidel
550	3.06	2.31
2018	49.2	12.01

4650	339.19	43.82
9396	1907.6	209.08

Table 5.4: Comparison of the Times to Convergence for Different Pressure Correction Solution Methods

5.4 Flow over a Backward Facing Step

The phenomena of flow separation in internal flows caused by sudden changes in geometry are well known. The importance of such flows to practical engineering applications dictates that any useful numerical algorithm must be able to calculate accurately the physical properties of this type of flow. A test case that has been used throughout the open literature to assess the accuracy of various numerical algorithms at calculating regions of separation in internal flows is the backward facing step. The test case consists of fluid flowing down a small inlet channel, width h, and then passing over a step, height S, into a wider channel. A region of separation forms behind the step that reattaches further downstream, a schematic of the geometry is given in Figure 5.13. The geometry of this test case is quite simple. Nevertheless the resulting flow pattern is representative of separation regions found in much more complicated domains and therefore has been proven to be a very useful test case.

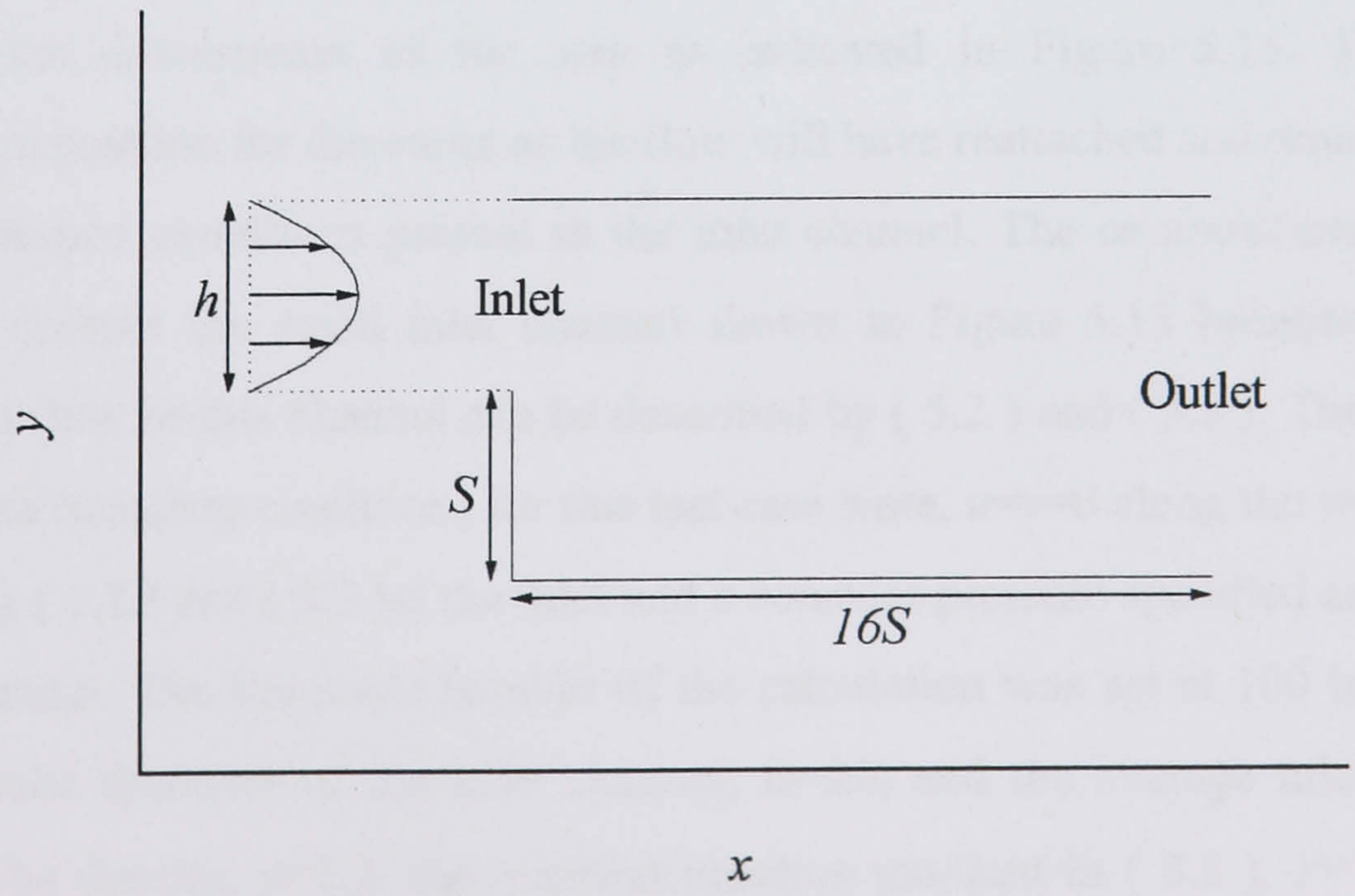


Figure 5.13: Schematic of the Backward Facing Step

Armaly *et al.*[87] along with many others [88,89,90,91] published both theoretical and experimental data for the backward facing step test case. Here, Armaly *et al.*'s[87] experimental data, measured using laser-Doppler anenometry, is used for comparison purposes. Testing the results of the solution algorithm against experimental as well as calculated benchmark data ensures that the algorithm produces results that are consistent with real fluid flows not just theoretical calculations. Also theoretical results may suffer from inaccuracies that it would be desirable to remove from a new algorithm and these can only be highlighted if experimental data is used as a benchmark.

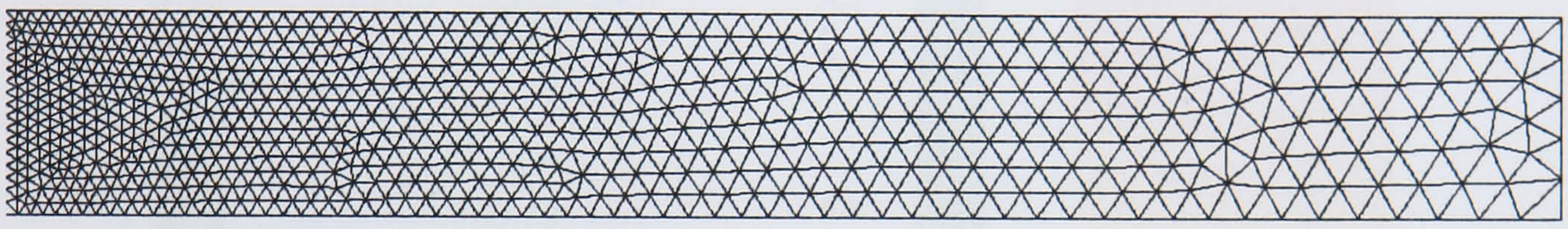


Figure 5.14: 1452 CV Backward Facing Step Mesh

Three grids were used to calculate solutions to the backward facing step test case. They contain 1452 CV, 5165 CV and 10704 CV the coarsest mesh is shown in Figure 5.14. The control volumes of each of the grids are packed densely in the region where the separation is expected to form and gradually become coarser as the outlet is approached and the flow pattern becomes simpler. The outlet is set sixteen step heights downstream of the step as indicated in Figure 5.13. This is an appropriate position for the outlet as the flow will have reattached and returned to the fully developed conditions present in the inlet channel. The computational domain does not contain the small inlet channel shown in Figure 5.13 because the fully developed flow in this channel can be described by (5.2) and (5.3). Therefore the appropriate boundary conditions for this test case were, u=v=0 along the walls, u and v given by (5.2) and (5.3)at the inlet and a constant pressure specified at the outlet of the domain. The Reynolds number of the calculation was set at 100 based upon the hydraulic diameter of the inlet channel, D=2h, and the average inlet velocity, U_{avg}=1. The density, ρ=1.2, the constant pressure gradient in (5.2), P=1, and the pressure at the outlet was set equal to zero. The expansion ratio of the test case was 1:1.94 which results in h=1.06 if S=1. Each calculation performed was iterated until the residual of each of the governing equations was reduced to 10^{-4}.

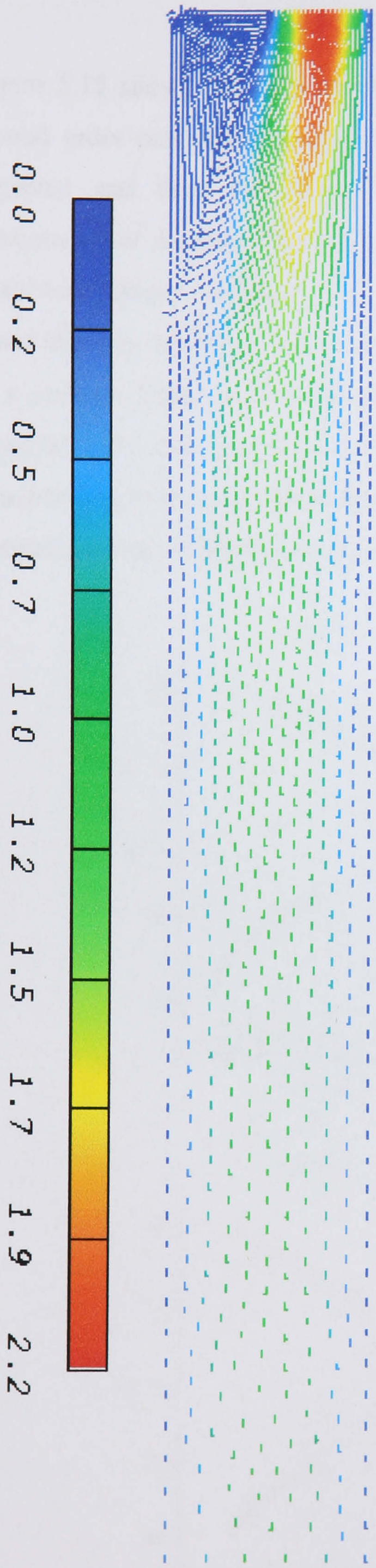


Figure 5.15: Velocity Vectors over the Backward Facing Step Coloured by their Magnitude (m/s)

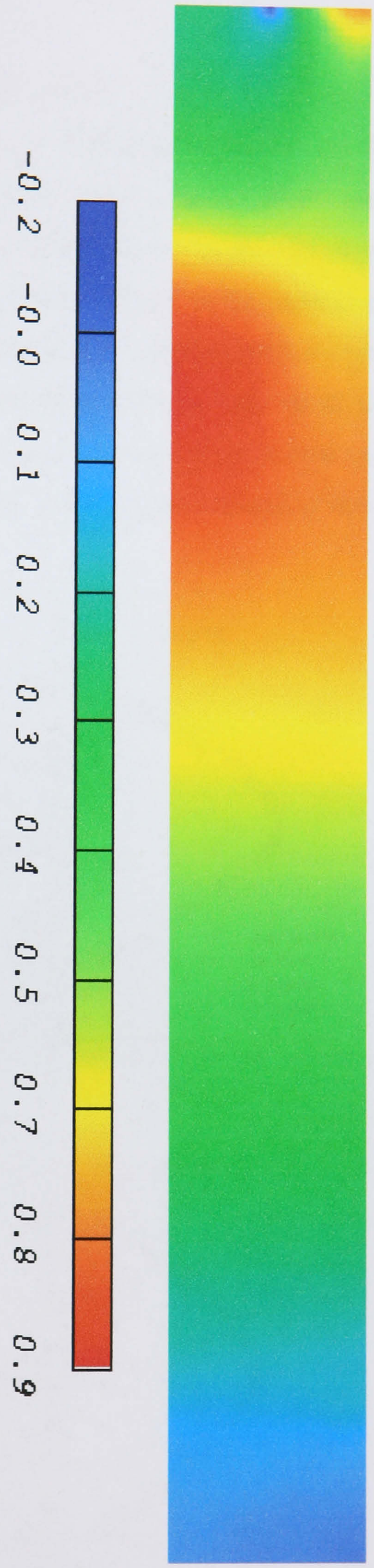
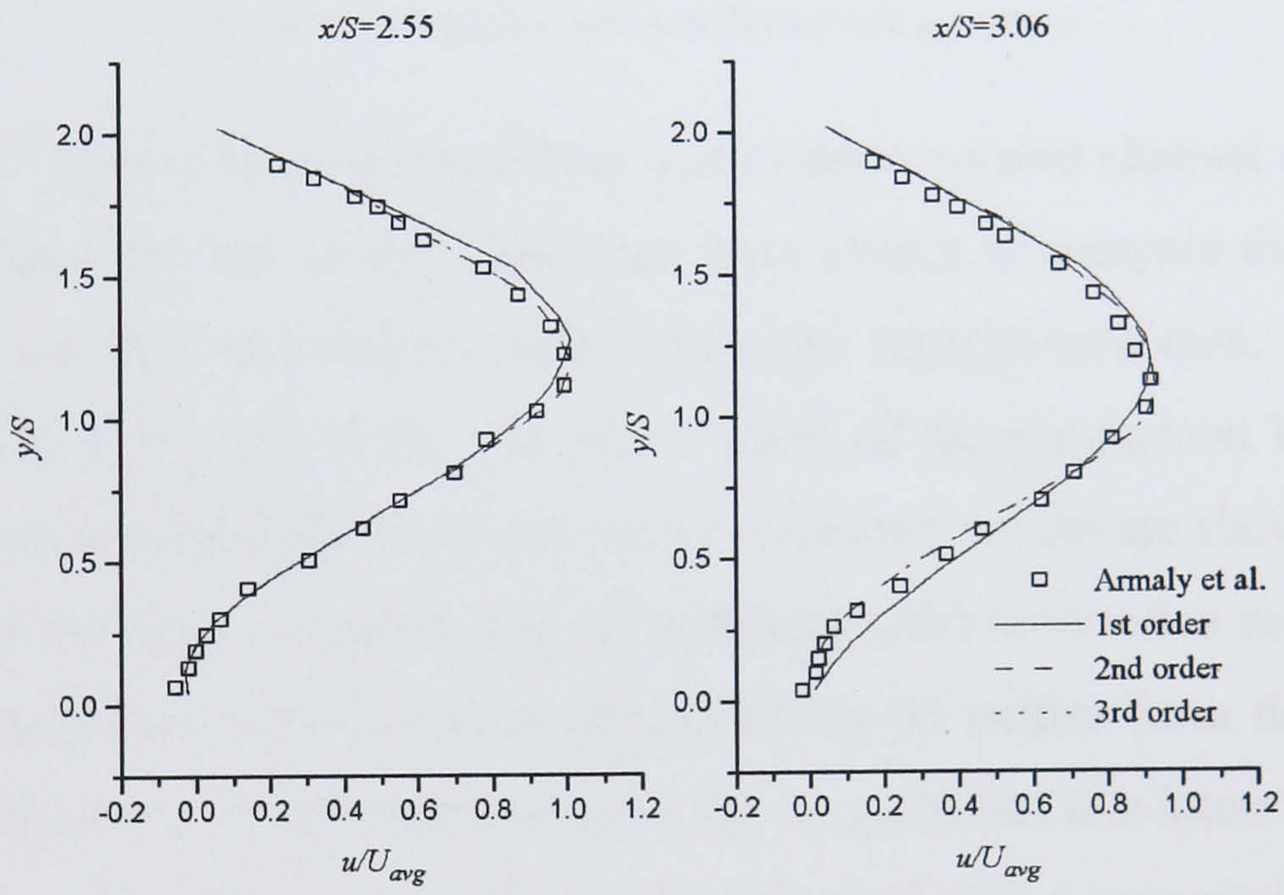


Figure 5.16: Relative Pressure (Pa) Contours over the Backward Facing Step

Figure 5.15 shows the velocity vectors calculated on the 5165 CV grid using the second order convective model. A region of separation forms behind the step as expected and then the flow reattaches at approximately three step heights downstream of the step. This pattern is also clearly shown in the calculated pressure distribution shown in Figure 5.16. An area of low pressure in the region immediately below the step causes the separation and then gradually the pressure contours change to a constant linear pressure gradient, consistent with the fully developed profile expected after the flow has reattached. It should also be noted that no spurious pressure distributions are found at the outlet of the domain because of the use of the constant pressure boundary condition.



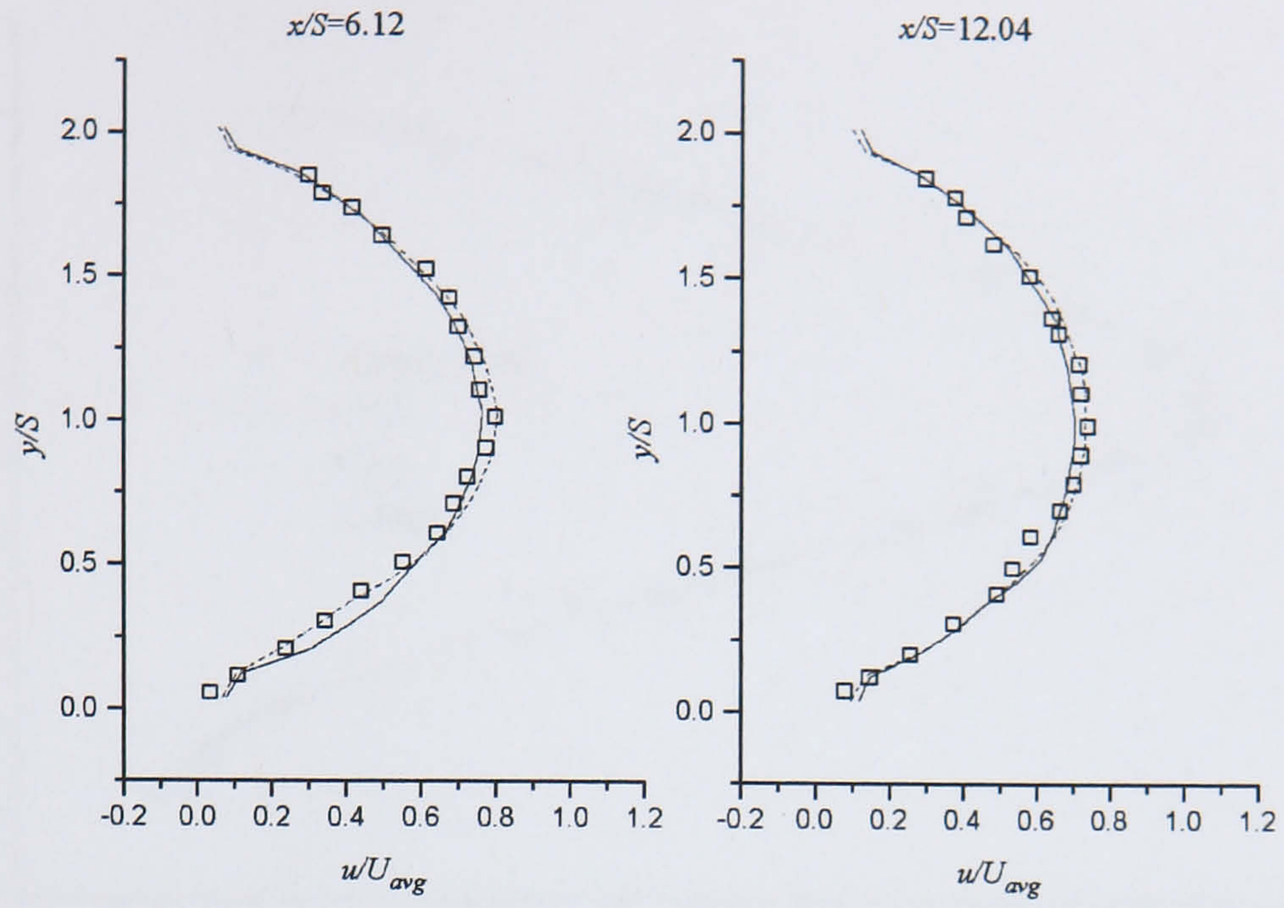


Figure 5.17: Velocity Profiles at Various x/S Locations

Armaly *et al.*[87] measured velocity profiles across the expanded channel at a number of different locations. Six of these locations were chosen to compare the calculated results from the three convective models with the experimental data. These were x/S=2.55, 3.06, 3.57, 4.80, 6.12, and 12.04. Plots of the comparison between the calculated results on the 5165 CV mesh and experimental results are shown in Figure 5.17. As with the skew cavity the second and third order convective models match the experimental data well, whilst the first order model suffers from the effects of numerical diffusion in the presence of high velocity gradients. It is interesting to note that at Re=100 in the skew cavity all models performed with a very similar accuracy whilst here the differences between the first order model and the second and third order schemes are greater. This behaviour indicates that the inaccuracies of first order modelling are problem dependent and that this convective model cannot necessarily be expected to perform well even at low Reynolds number. Figure 5.18 shows a comparison of the velocity profile calculated using second order convective modelling at x/S=2.55 for the three different grid sizes. Clearly a grid independent solution has been reached, as the solutions for the two finest grids are virtually identical. A comparison of the continuity equation convergence history for all three grids is shown in Figure 5.19. The second order convective model has been used in these calculations and as would be expected the number of iterations to convergence increases with grid size.

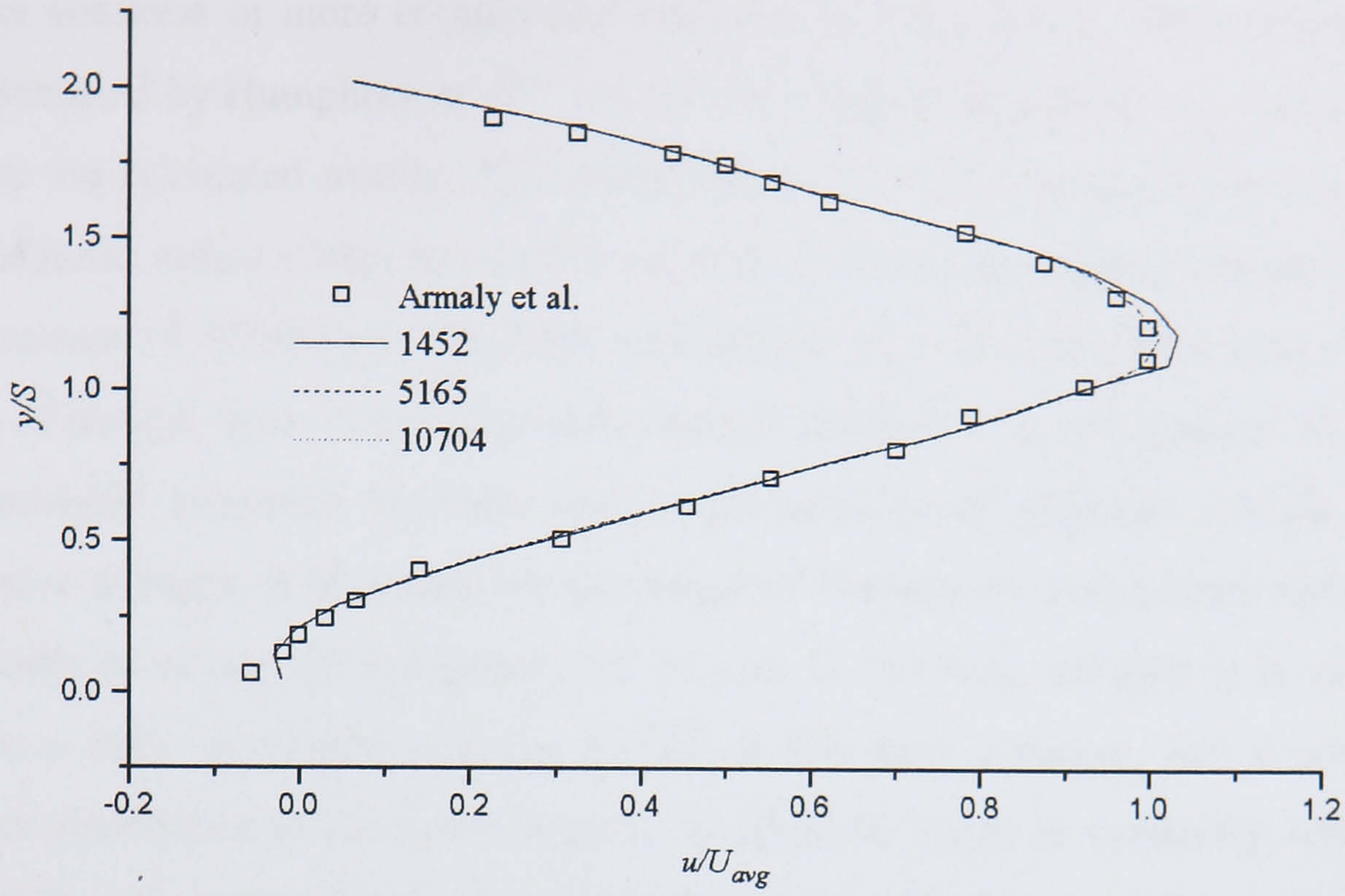


Figure 5.18: Variation of the x/S=2.55 Velocity Profile as a Function of Grid Fineness

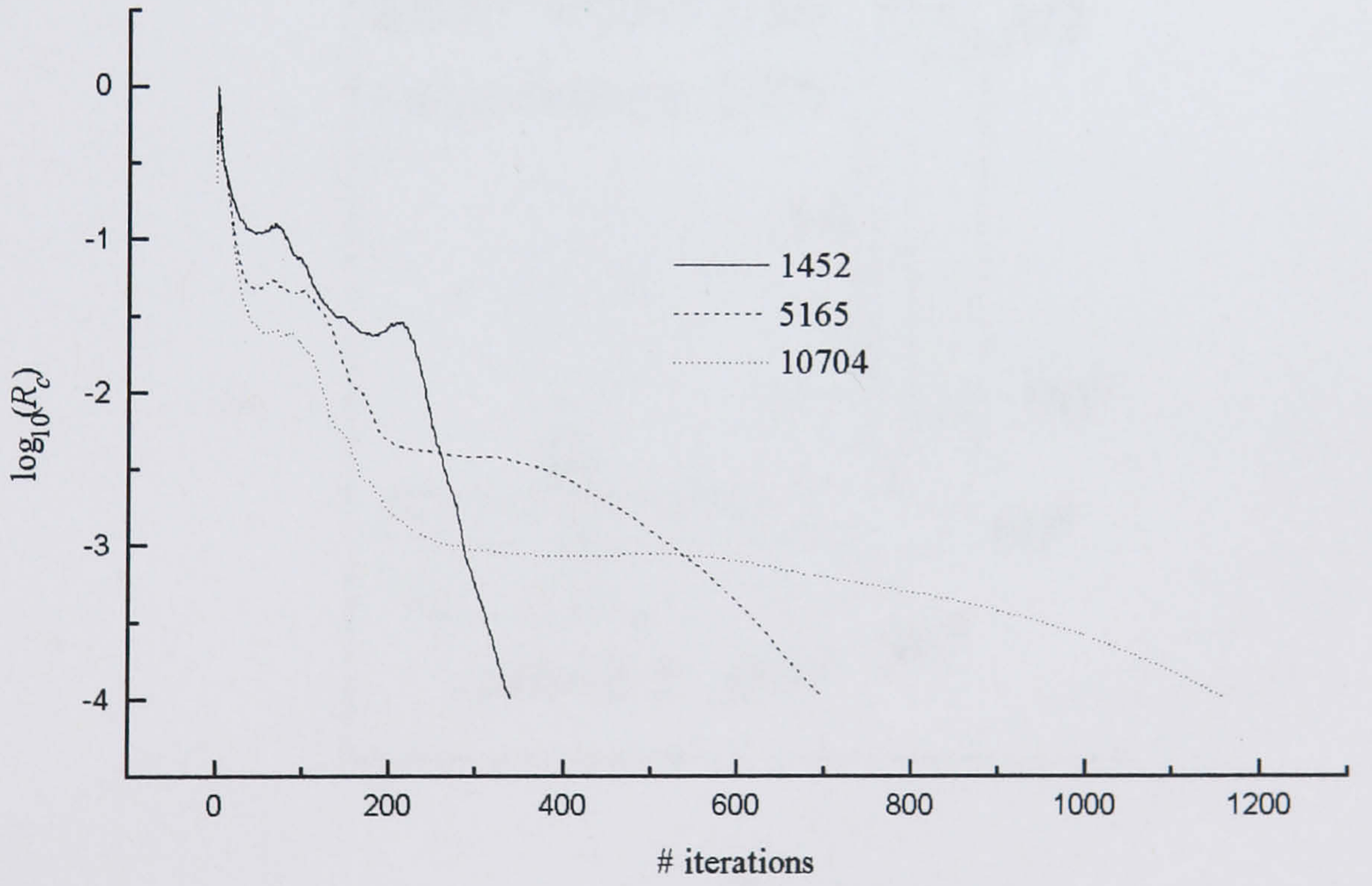


Figure 5.19: Backward Facing Step Convergence Histories for Different Mesh Sizes

5.5 Flow in a Square Duct with Strong Curvature

Laminar flow in a curved duct of square cross-section is a standard benchmark test case for three-dimensional internal flows[92,93,94]. As with the skew cavity and backward facing step this test case is a simple geometry that creates flow patterns

that are common in more complicated engineering applications. The experimental data measured by Humphrey *et al.*[93] using laser-Doppler anenometry is used here to validate the calculated results. The configuration of the experimental rig was a 90^0 bend of mean radius 92mm attached to the end of a 1.8m rectangular channel with a cross-section of 40x40mm. The bend was located in a vertical plane with a 1.2m length of straight duct of the same cross-section attached to its downstream end. For computational purposes the inlet and outlet sections of physical domain were reduced to a length of $5h$ where h is the length of one side of square cross-section. It is possible to reduce the computational domain in this way, because it is valid to assume a fully developed velocity profile at this inlet location, and a constant pressure distribution at the outlet position. In addition, owing to symmetry, only one half of the physical domain was modelled. Figure 5.20 shows a schematic of the computational domain described above.

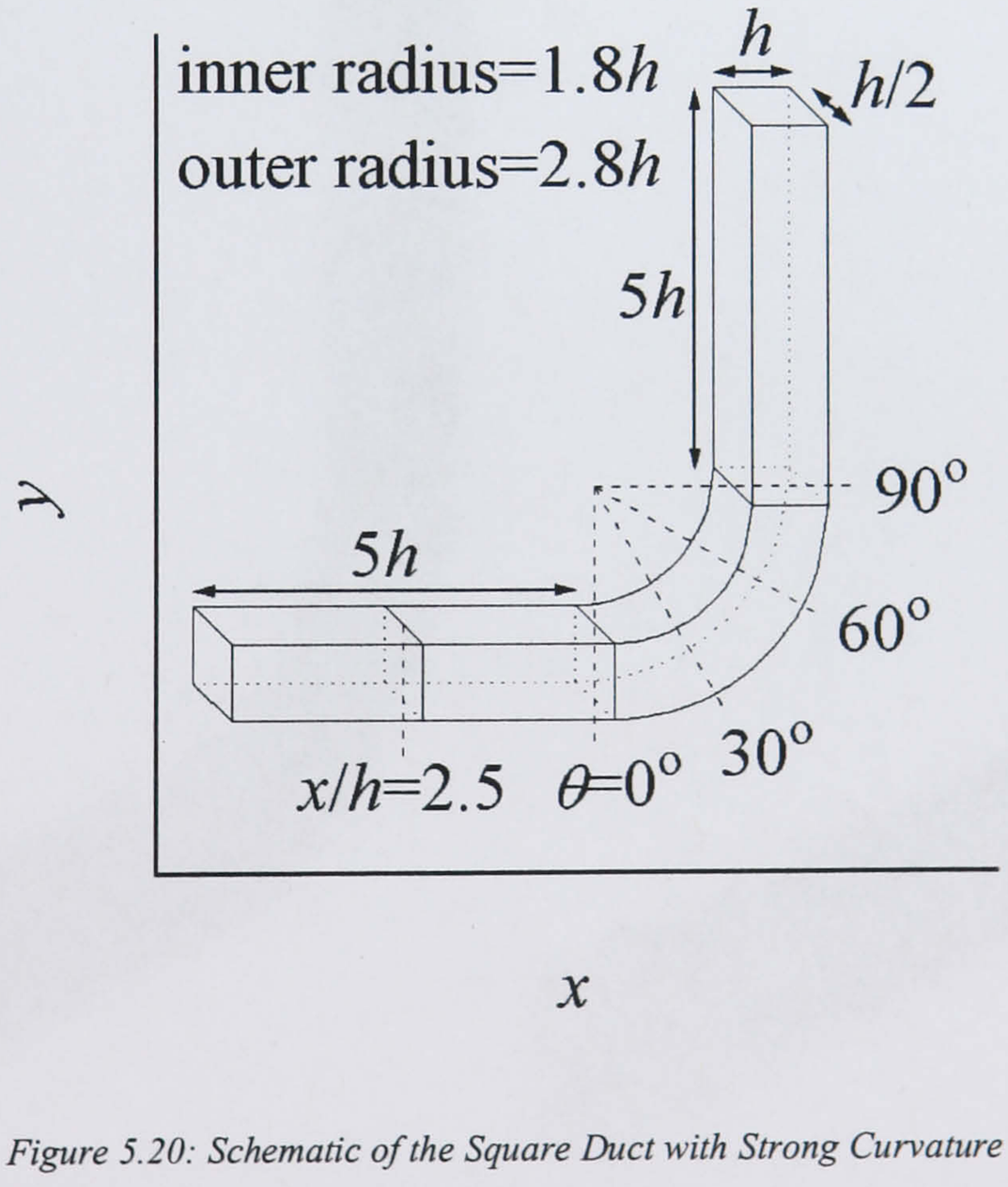


Figure 5.20: Schematic of the Square Duct with Strong Curvature

The calculations were performed at a Reynolds number of 790 based upon the hydraulic diameter, $d=h$, and the bulk velocity, U_b. The fully developed velocity profile specified at inlet is given by the following expressions

$$u(y,z) = -\frac{4h^2}{\mu\pi^3}\sum_{i=1,3,5,\ldots}^{\infty}(-1)^{\frac{i-1}{2}}\left[1-\frac{\cosh(i\pi z/h)}{\cosh(i\pi h/2h)}\right]\frac{\cos(i\pi y/h)}{i^3} \quad (5.4)$$

$$v = 0 \quad (5.5)$$

$$w = 0 \quad (5.6)$$

where $-h/2 \le y \le h/2$ and $-h/2 \le z \le h/2$. The other boundary conditions applied were $u=v=w=0$ along the walls of the duct, zero normal velocity through the symmetry boundary and a constant pressure of zero at the outlet. All calculations were iterated until the residual of each of the governing equations was reduced to 10^{-4}.

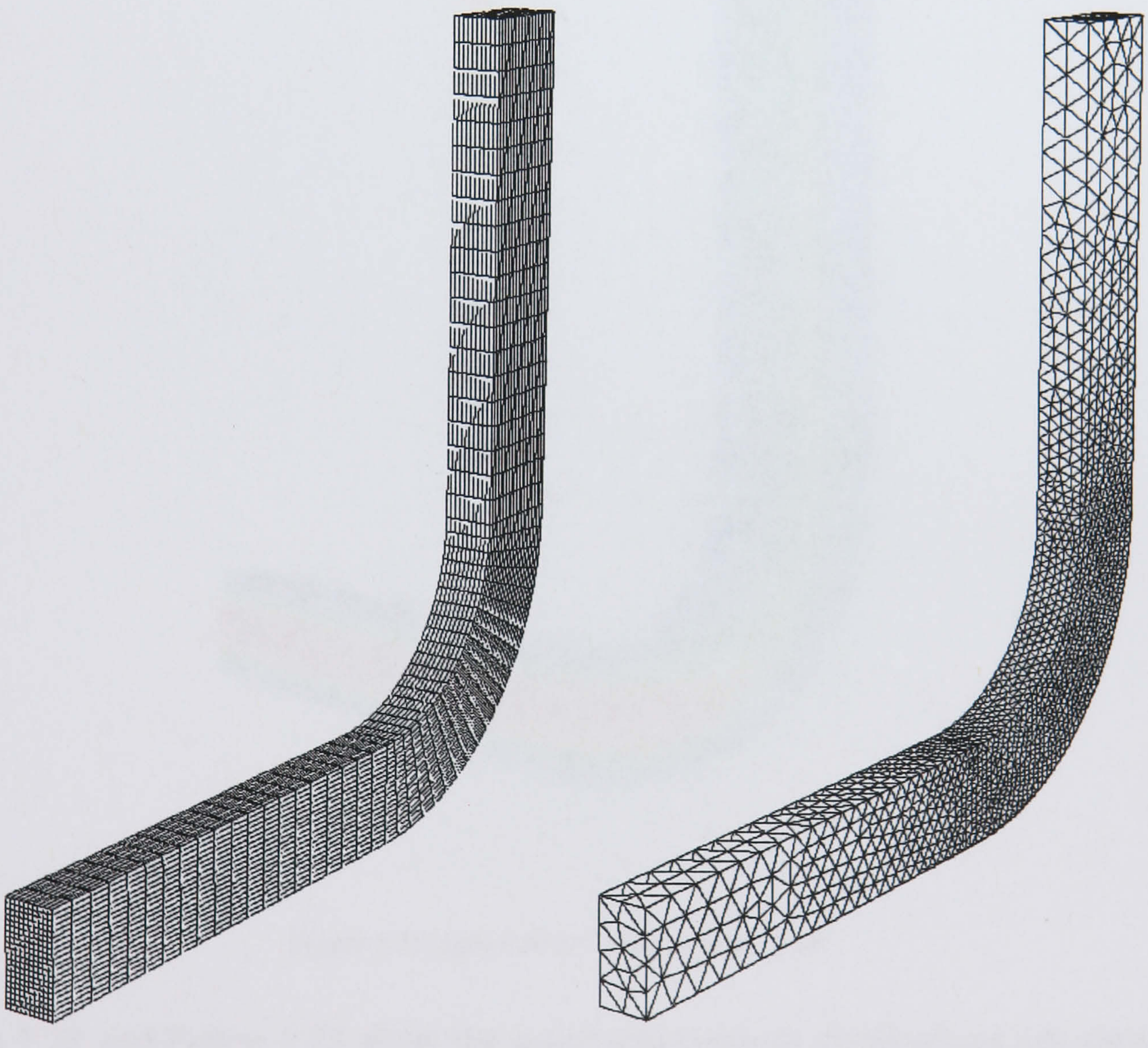


Figure 5.21: Structured and Unstructured Duct Meshes

Six meshes were used to calculate solutions, three unstructured tetrahedral grids and three structured hexahedral grids. The tetrahedral meshes contained 49193 CV, 116415 CV and 148835 CV whilst the hexahedral meshes consist of 20x10x60 CV, 28x14x84 CV and 40x20x120 CV. An example of each type of grid is shown in

Figure 5.21. The hexahedral grid has uniformly spaced control volumes whilst the tetrahedral grid is refined in the curved section of the grid. This concentration of control volumes in the tetrahedral mesh is intended to give the calculated solution greater resolution in the region where the most complicated flow patterns are likely to occur. Both structured and unstructured grids were used in order that a comparison could be made between the accuracy of the solutions calculated with each type of grid.

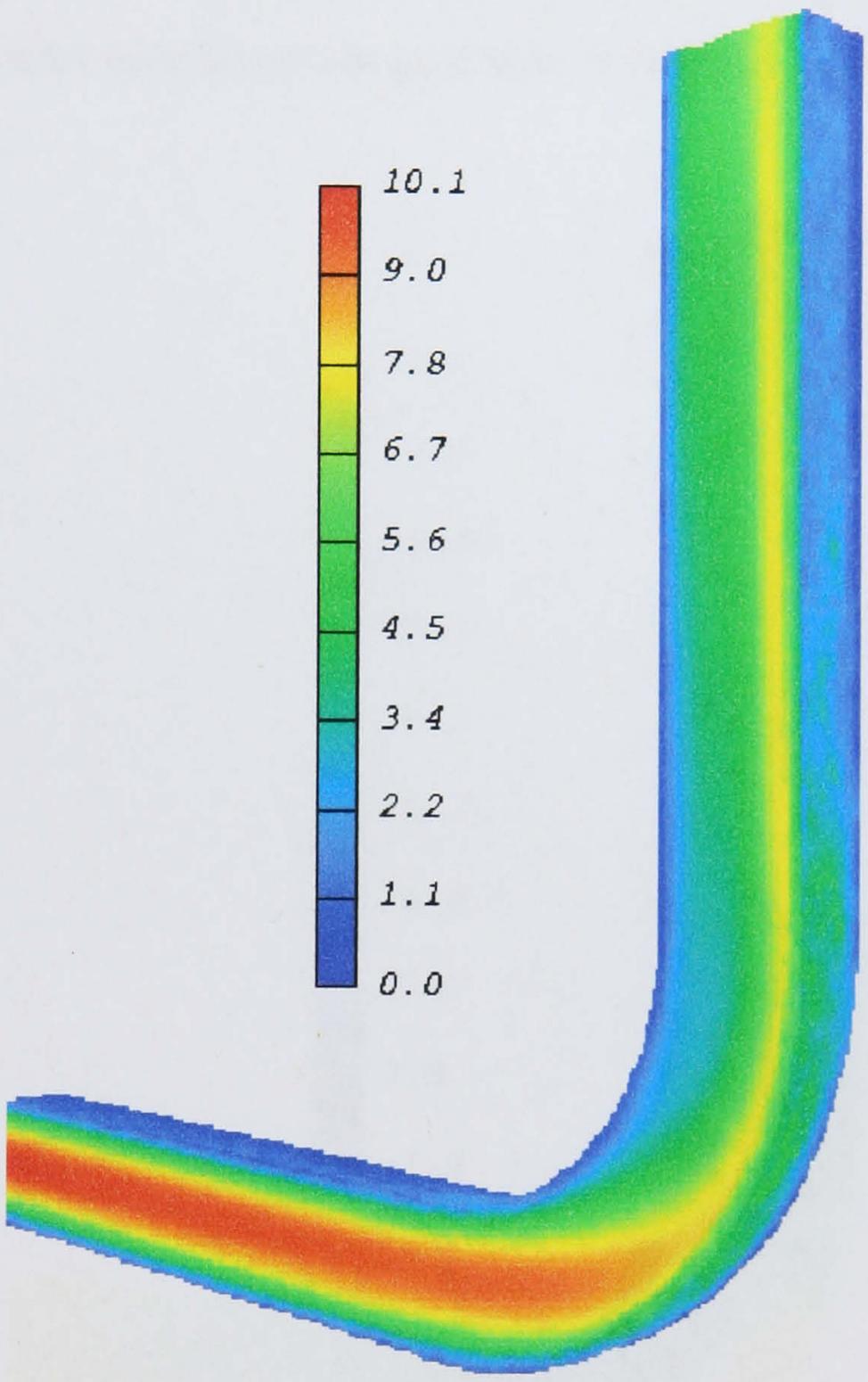


Figure 5.22: Speed (m/s) Contours in the Duct

Figure 5.22 and Figure 5.23 show the speed and pressure distributions calculated in the duct using the 148835 CV tetrahedral grid and second order convective modelling. The speed contour plot shows the fully developed velocity profile specified at inlet is maintained in the duct until the curved section begins. At this point there is a general movement of fluid away from the inner-radius wall and towards the outer-radius wall. This velocity profile is maintained after the curved

section of the duct finishes but as the outlet approaches fluid has begun to move back from the outer-radius wall to the centre of the duct. The pressure contour plot in the inlet straight duct section shows an essentially linear distribution in the streamwise direction as would be expected. When the fluid enters the 90^0 bend there is a favourable and adverse longitudinal pressure gradient at the outer and inner wall respectively. This, in addition to the centrifugal forces in the duct, causes the movement of fluid from the inner to the outer wall of the duct. As the fluid exits the curved section of the duct the pressure distribution once again settles into a linear distribution in the streamwise direction as it was in the inlet section.

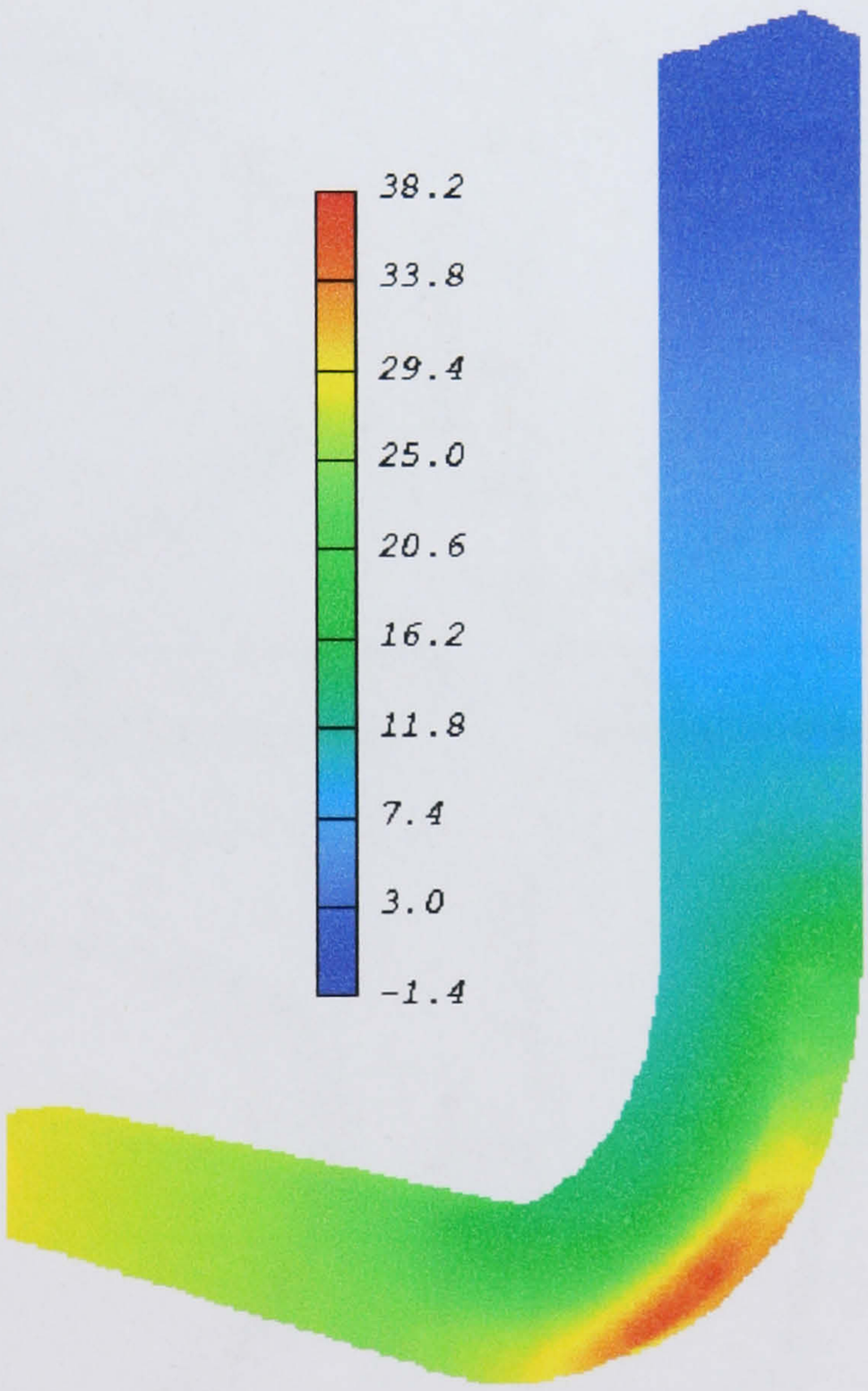
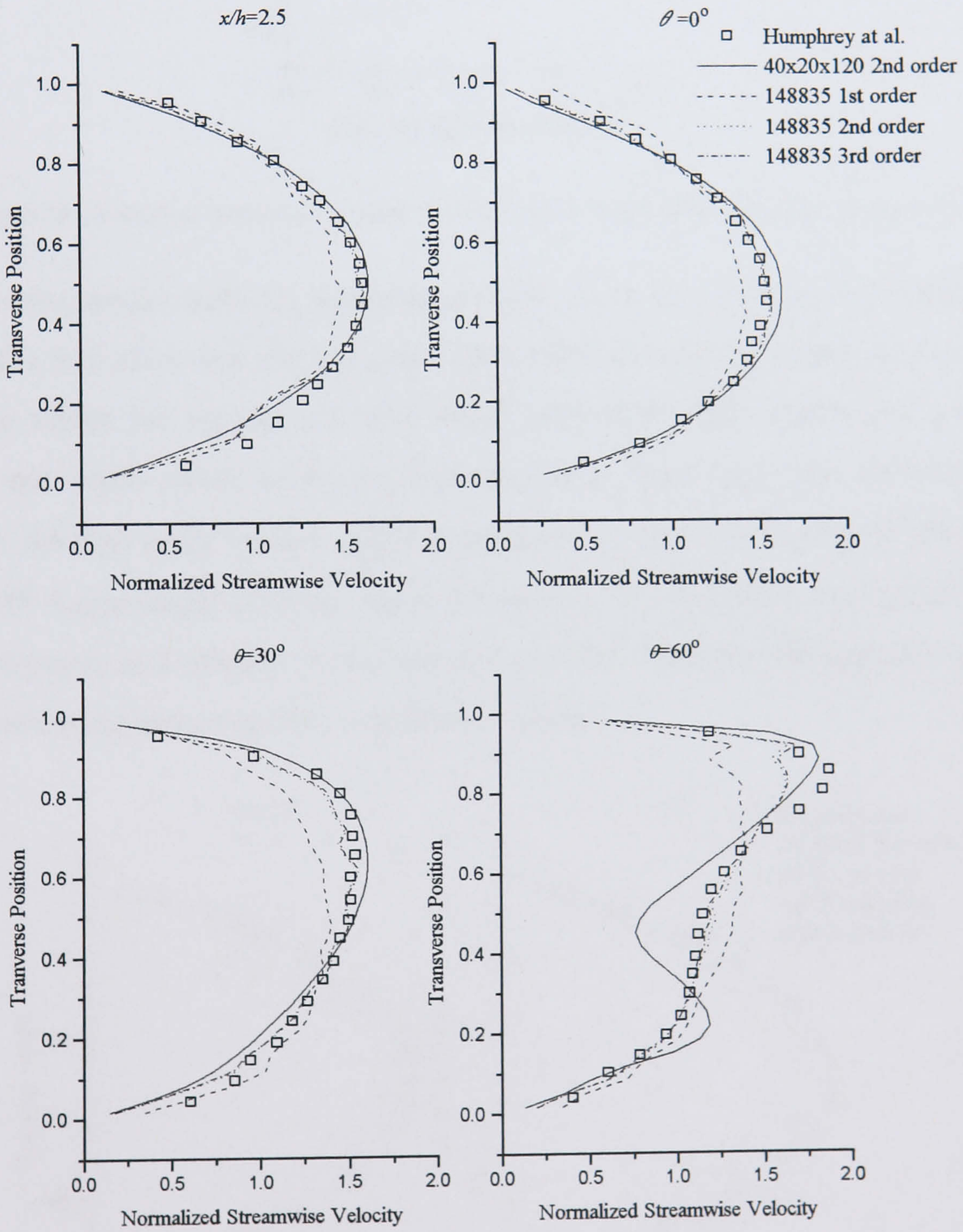


Figure 5.23: Relative Pressure (Pa) Contours in the Duct

Humphrey et al.[93] measured the velocity in the duct at five longitudinal locations, x/h=2.5, θ=0^0, θ=30^0, θ=60^0 and θ=90^0, all shown in Figure 5.20. It is these measurements that are used here to compare the three convective models and the results from the structured and unstructured grids. Velocity profiles at each of the

longitudinal locations are plotted in Figure 5.24 and Figure 5.25, along two lines z/h=0.25 and z/h=0 (centreline). The profiles presented are calculated using the 148835 CV unstructured grid and the 40x20x120 CV structured mesh. For the unstructured mesh, results are presented using first, second and third order convective modelling. In the case of the structured grid, only results using the second order convective model are presented as the performance of the various convective models is identical to those of the unstructured grid.



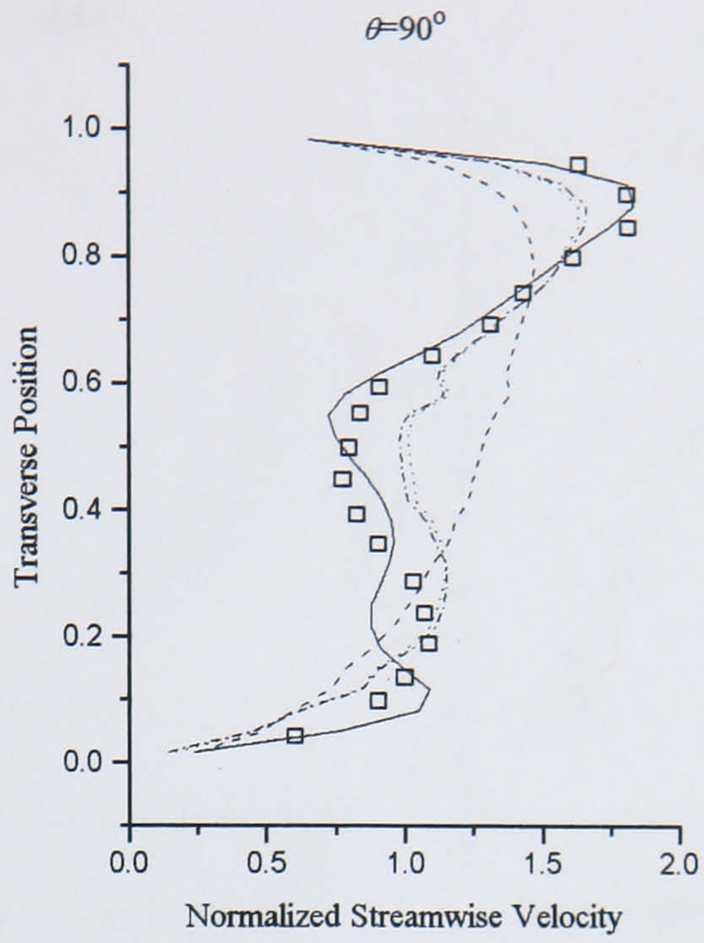
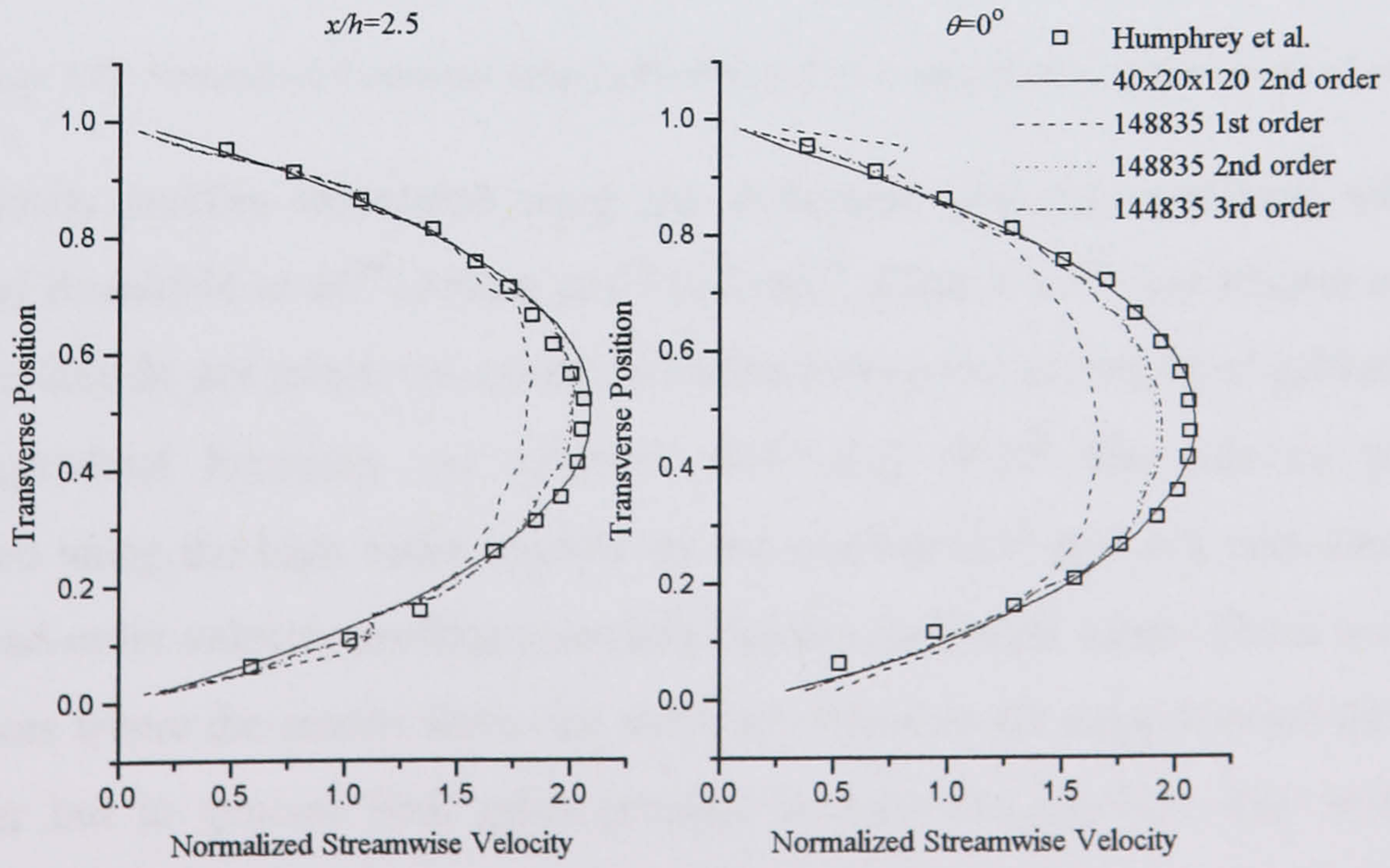


Figure 5.24: Normalized Streamwise Velocity Profiles at Five Longitudinal Locations along z/h=0.25

Consider the results from the unstructured grid. As in two dimensions Figure 5.24 and Figure 5.25 show that the first order model suffers from the effects of numerical diffusion whilst the second and third order models produce results that are very similar and much closer to the experimental data. There are some discrepancies between the high order models and the experimental data particularly at the θ=60^0 and θ=90^0 longitudinal locations. Here, the models do not predict the magnitude of all the maxima and minima in the velocity profiles correctly although the general pattern of the experimental data is certainly evident.



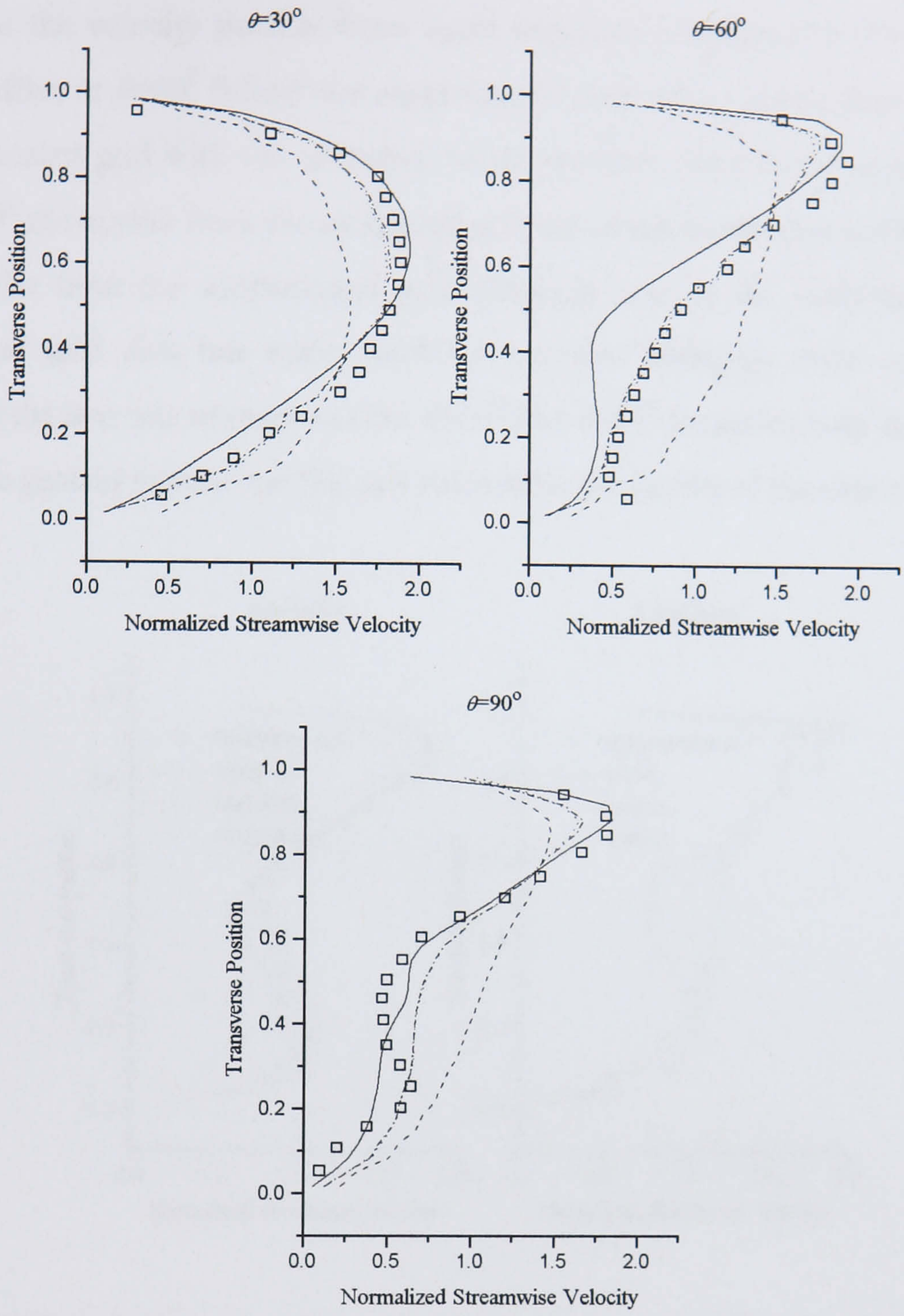


Figure 5.25: Normalized Streamwise Velocity Profiles at Five Longitudinal Locations along z/h=0

The velocity profiles calculated using the structured grid are consistent with the results of Rosenfeld *et al.*[94], Jessee and Fiveland[95], Choi *et al.*[96] and Rogers *et al.*[97]. However they do not match the results calculated using the unstructured grid at all of the longitudinal locations. At x/h=2.5, θ=0^0 and θ=30^0 the velocity profiles calculated using the high order models on the unstructured grid are very similar to the second order velocity profiles calculated on the structured mesh. There are a few differences where the results from one mesh are closer to the experimental data than the other but in general both grids produce comparable profiles. The structured profiles are in general a lot smoother but this is a product of the regular position of the cell nodes rather than an indication of greater accuracy. At the θ=60^0 and θ=90^0

locations the velocity profiles have some significant differences. The unstructured grid profiles at θ=60^0 follow the experimental data more closely than the results of the structured grid with the exception of the extrema close to the outer-radius wall. At θ=90^0 the results from the structured grid are closer to the experimental data than the results from the unstructured grid although near to the inner-radius wall the structured grid data has some spurious extrema. Although there are differences between the two sets of results at the θ=60^0 and θ=90^0 locations both data sets follow the same general pattern and this is a reasonable prediction of the experimental data.

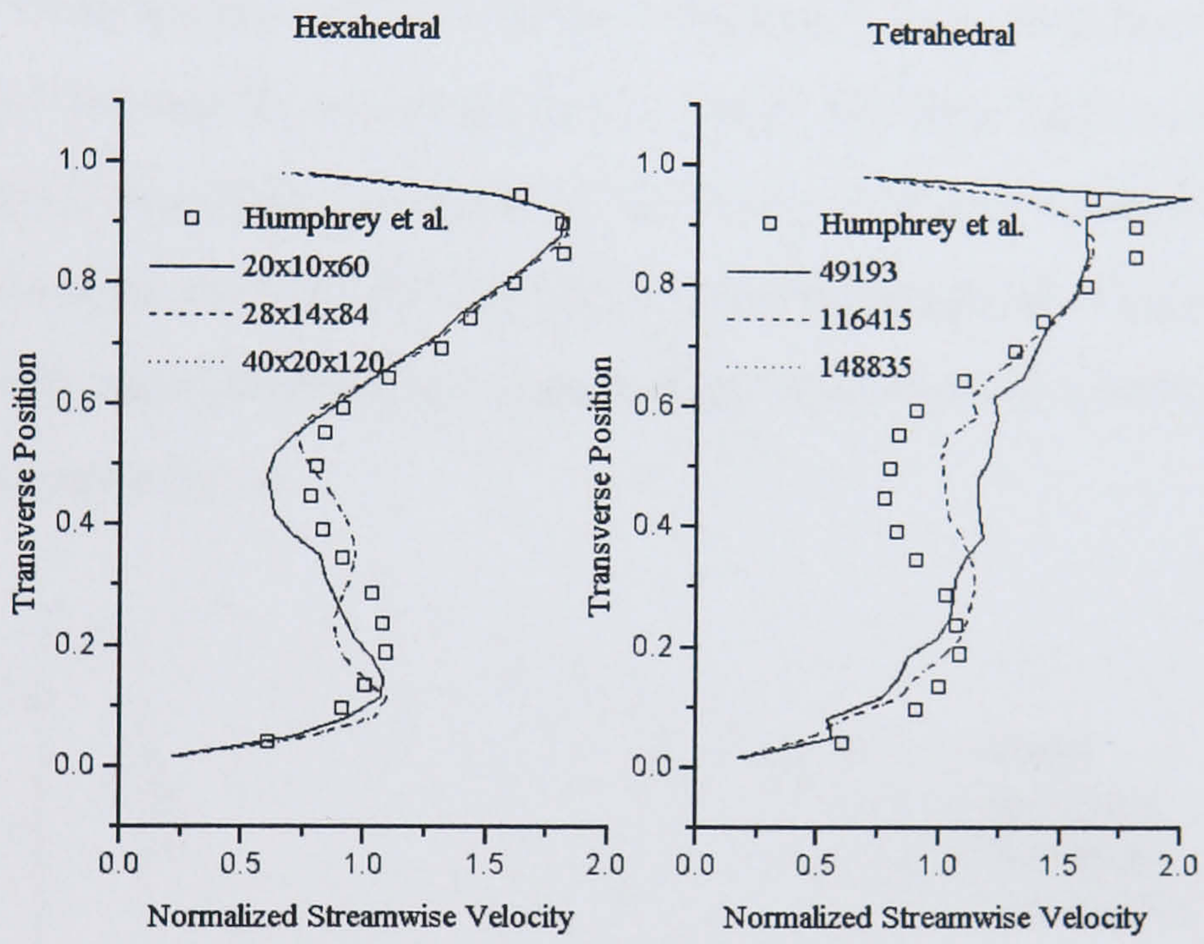
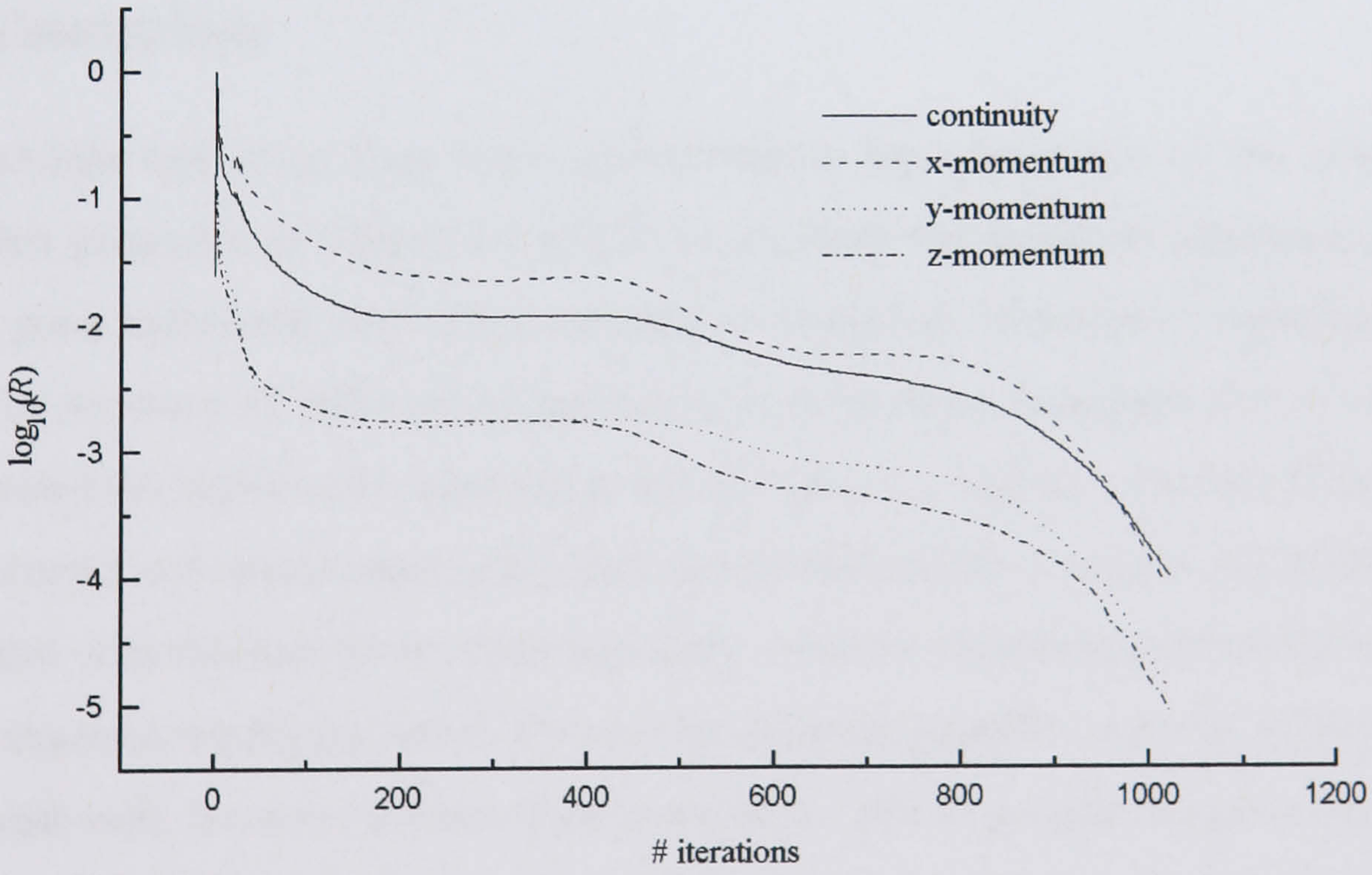


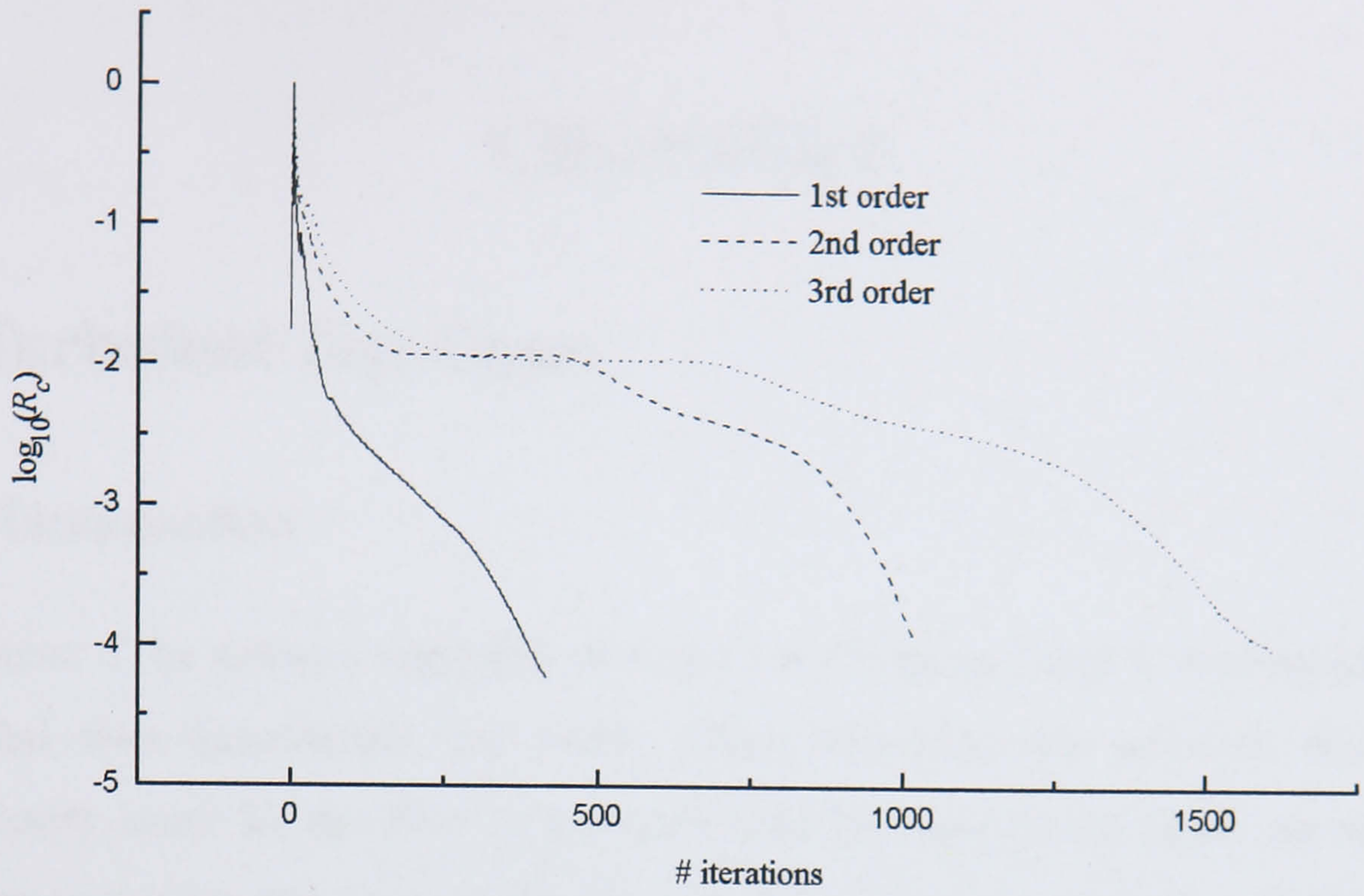
Figure 5.26: Normalized Streamwise Velocity Profile at θ=90^0 along z/h=0.25 as a Function of Grid Fineness

The unstructured and structured grids used for comparison in Figure 5.24 and Figure 5.25 are clearly of a different size. It could then be possible that the differences between the two sets of data are caused by grid dependence. However, Figure 5.26 shows the velocity profile at θ=90^0 along z/h=0.25 as a function of grid fineness, for both the unstructured and structured meshes, and clearly the solutions calculated using the finest grid in both cases are grid independent. It is therefore inevitable that the grids cause the differences between the solutions. This is possible because the calculation of gradients can be grid dependent and therefore terms in the governing equations such as the convective and non-orthogonal diffusion term will be effected.

To conclude this test case Figure 5.27 show plots of the convergence histories of all four governing equations and a comparison of the continuity equation convergence for the three convective models. The convergence histories in Figure 5.27(a) are from the case calculated using second order convective modelling and 148835 CV grid. They show a very similar convergence pattern for each of the governing equations with the continuity equation ultimately taking the greatest number of iterations to converge. The comparison of the convergence of three convective models in Figure 5.27(b) is also from the calculations using the 148835 CV grid. Interestingly in this three-dimensional case with each increase in the order of accuracy of the convective model there is an increase in the number iterations it takes the algorithm to converge. This was not the case in two dimensions where the third and second order models both converged in a similar number of iterations. However in three dimensions it is likely that the third order modelling makes a more significant impact on how explicit the momentum equations are and hence there will be an increase in the number of iterations to convergence.



(a)



(b)

Figure 5.27: Duct Convergence Histories (a) Comparison between Governing Equations (b) Comparison between Convective Models

5.6 Conclusions

Four laminar test cases have been used to assess the capabilities of the solution algorithm presented in Chapters 3 and 4. In all cases the solutions calculated have shown good agreement with either benchmark numerical solutions or experimental data. The accuracy of different convective models has been compared and as would be expected the higher order convective models have the highest accuracy. Solutions on structured and unstructured grids have shown comparable accuracy, the problems associated with the derivative outlet boundary condition demonstrated and the effect of the algebraic multigrid tested. Overall the solution algorithm presented here has performed well, however laminar flow patterns are not in general found in the real world, and therefore Chapter 6 will continue the testing process, by assessing the accuracy of the algorithm at modelling turbulent flow.

CHAPTER 6

6 Turbulent Test Cases

6.1 Introduction

In Chapter 5 the solution algorithm developed in Chapters 3 and 4, was tested using two and three-dimensional test cases, which had Reynolds numbers that were sufficiently small for the flow to be considered laminar. As an initial set of trials laminar test cases are very useful because they allow the accuracy of the basic discretisation of the governing equations to be tested, without the additional complexity of introducing a turbulence model. However as the majority of flow patterns of practical interest have Reynolds numbers that are well in excess of the point where laminar flow becomes turbulent, it is essential that any useful solution algorithm must also be able to accurately model turbulent flow.

In this chapter three test cases are used to test the accuracy of the solution algorithm developed here at modelling turbulent flow. These test cases are turbulent flow over a NACA 0012 airfoil, turbulent flow over a backward facing step and turbulent flow through a square duct with strong curvature. Clearly, the second and third of these test cases are the turbulent counterparts of laminar test cases already presented in Chapter 5. These cases are repeated because as in laminar flow the flow patterns that result from these relatively simple geometries are representative of flow patterns that occur in much more complicated domains. The first test case is included because it is an example of an external flow, in contrast to all the other laminar and turbulent test cases that are exclusively internal flows.

Each of the three turbulent test cases are compared against experimental data measured for a variety of the dependent variables. The accuracy of the first, second and third order convective models are tested to determine whether the benefits of higher order modelling shown in laminar flow are maintained when flows become turbulent. Also the effect of the wall function approximation is considered and the

merits of adopting a low Reynolds number version of the k-ε model discussed. The influence that the introduction of turbulence modelling has upon the convergence characteristics of the various convective schemes is also considered as is the effect grid refinement has on the accuracy of the algorithm. The three test cases chosen allow all the common boundary conditions to be tested to ensure they are correctly implemented and appropriate to the boundaries they are prescribed along. As in Chapter 5 the test cases chosen here offer sufficiently complicated flow patterns, to test effectively the ability of the solution algorithm to accurately model turbulent flows in a variety of domains.

6.2 Turbulent Flow over a NACA 0012 Airfoil

The design of aircraft wings and turbomachinery components over the past eighty years has resulted in the publication of a large amount of experimental data relating to the flow past wing sections. The majority of this data has been measured at Reynolds numbers that exceed the point at which fluid flow becomes turbulent and hence this pool of data can be used as a source of benchmark test cases to assess the accuracy of the turbulent solution algorithm presented here. The National Advisory Committee for Aeronautics (NACA) series of four digit airfoils is one example of a family of wing sections for which there is published high quality experimental data. The NACA 0012 wing section belongs to this family and has been used by a number other authors[20] as a benchmark test case and is used for that purpose here. Its geometry can be described by the following equation

$$\pm\frac{y}{c}=0.60\left(0.29690\sqrt{\frac{x}{c}}-0.12600\frac{x}{c}-0.35160\left(\frac{x}{c}\right)^2+0.28430\left(\frac{x}{c}\right)^3-0.10150\left(\frac{x}{c}\right)^4\right) \tag{6.1}$$

$$\frac{R_0}{c}=0.0158 \tag{6.2}$$

where y is the thickness of the wing section, x is the distance along the wings chord, c is the chord length and R_0 is the airfoil's leading edge radius. A schematic of the geometry is shown in Figure 6.1.

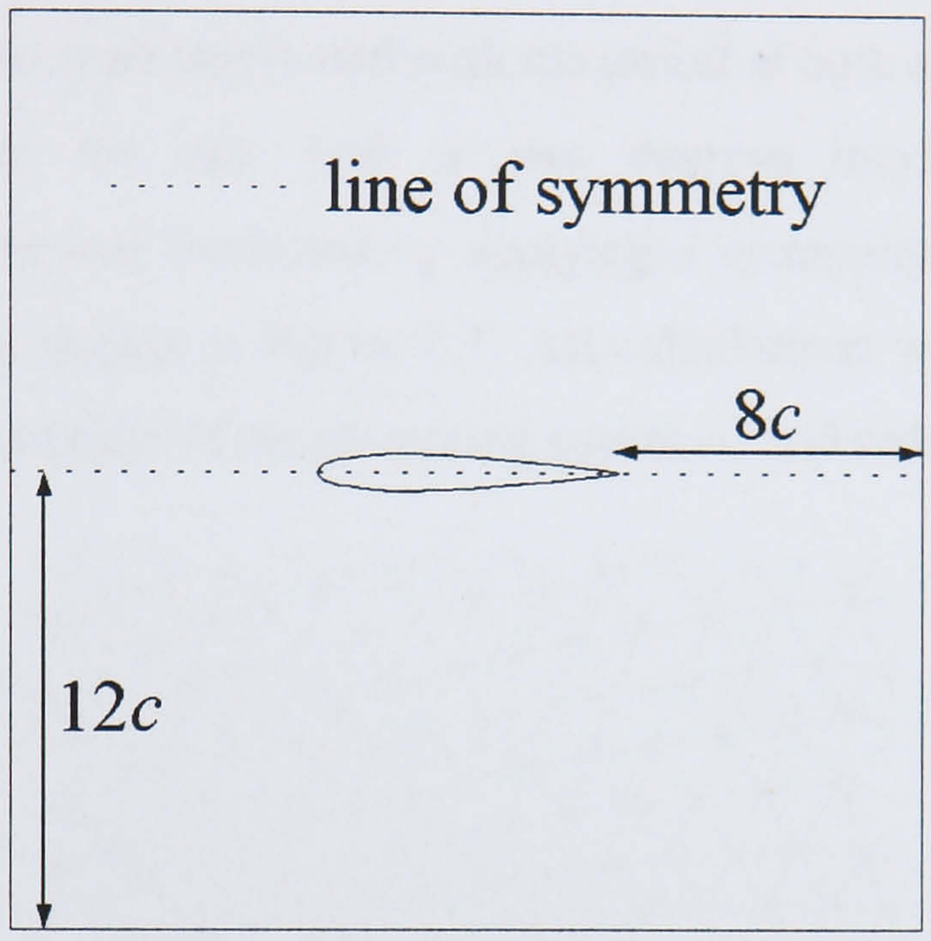
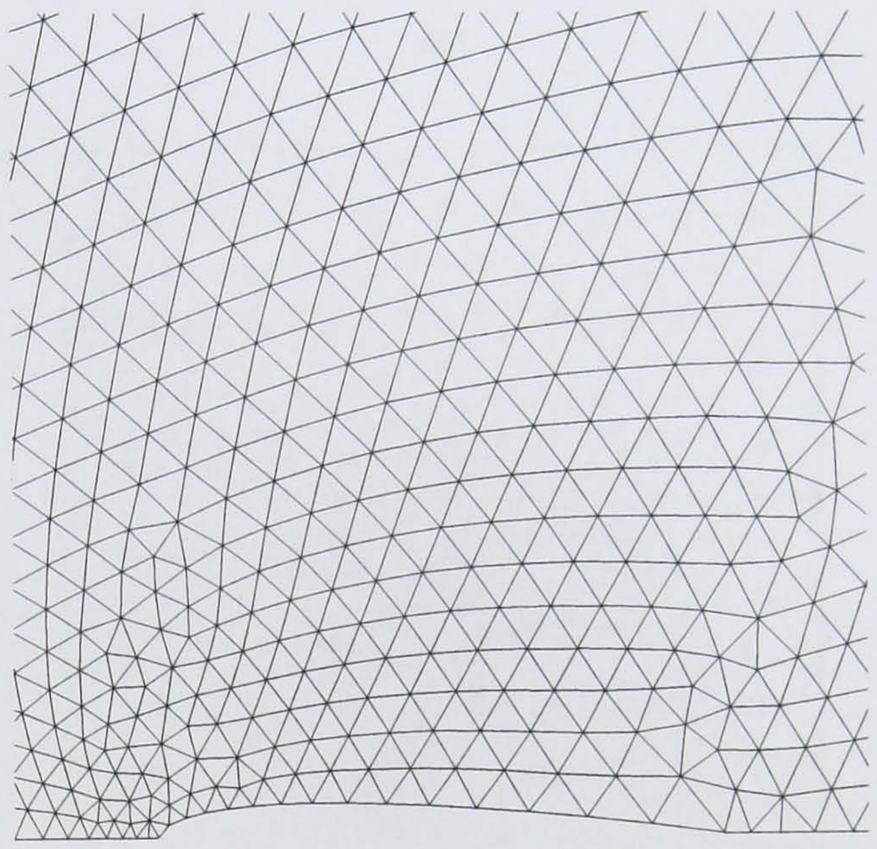


Figure 6.1: Schematic of the NACA 0012 Airfoil

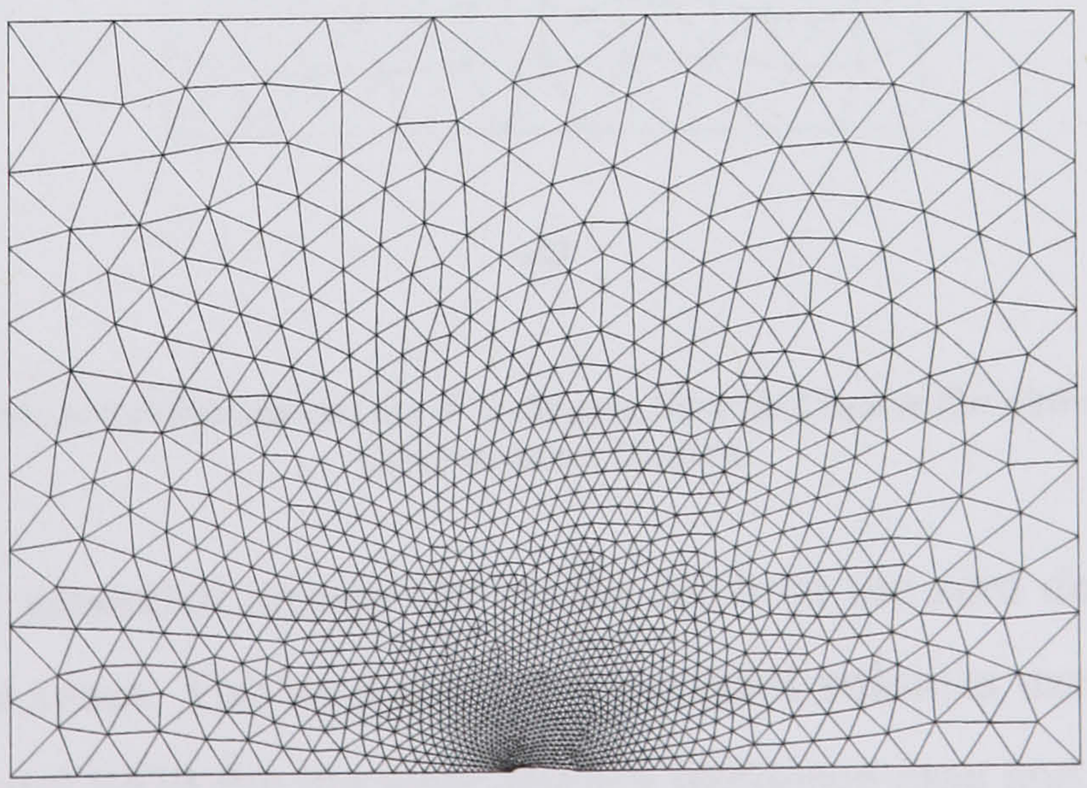
Gregory and O'Reilly[98] published the experimental data chosen to compare the numerical simulation against. Their work was performed in a low speed wind tunnel on a two-dimensional model of a NACA 0012 airfoil. The intention of Gregory and O'Reilly's[98] research was not only to measure the aerodynamic characteristics of the smooth airfoil but also to analyse the effect of upper surface roughness. However, here only the data measured over the smooth airfoil will be used for comparison purposes. The only flow field variable measured by Gregory and O'Reilly[98] was the surface static pressure, nevertheless this was measured around the complete airfoil and provides a good initial check of the accuracy of the algorithm.

The computational domain used in the numerical simulation can be seen in Figure 6.1. The inlet and outlet of the domain were located at eight chord lengths away from the leading edge and trailing edge respectively, whilst the top and bottom boundaries were placed at twelve chord lengths from the airfoil. The free stream velocity, U_∞, was set at 55 m/s along the inlet, top and bottom boundaries. The Reynolds number of the calculation based upon the chord length of the airfoil and the free stream velocity was equal to 2.8×10^6. The density, ρ, equalled 1.225 and the chord length, c,

was unity resulting in a viscosity, μ, of 2.40625×10^{-5}. Along the surface of the airfoil the no-slip condition was applied with standard wall functions allowing for the large velocity gradients and inadequacies of the k-ε turbulence model in the region close to the airfoil. At the outlet of the computational domain a constant pressure of zero was defined. Computations were performed with the airfoil at both a zero and six degrees angle of attack. For the case with a zero degrees incidence only half the computational domain was modelled by applying a symmetry condition along the appropriate boundary shown in Figure 6.1. All calculations were iterated until the normalized residuals of each of the governing equations had reduced to 10^{-3}.

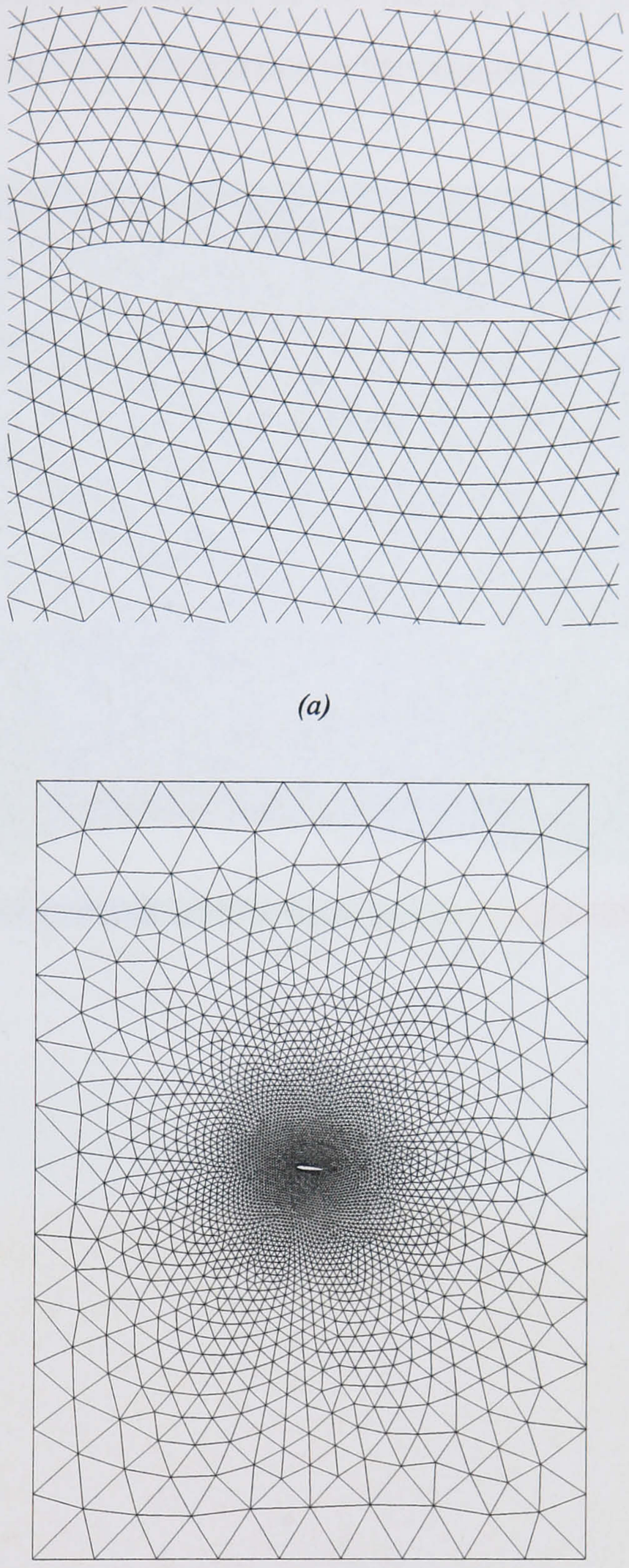


(a)



(b)

Figure 6.2: The Unstructured Grid Around the 0° Incidence Case (a) Close to the Airfoil (b) In the Complete Computational Domain



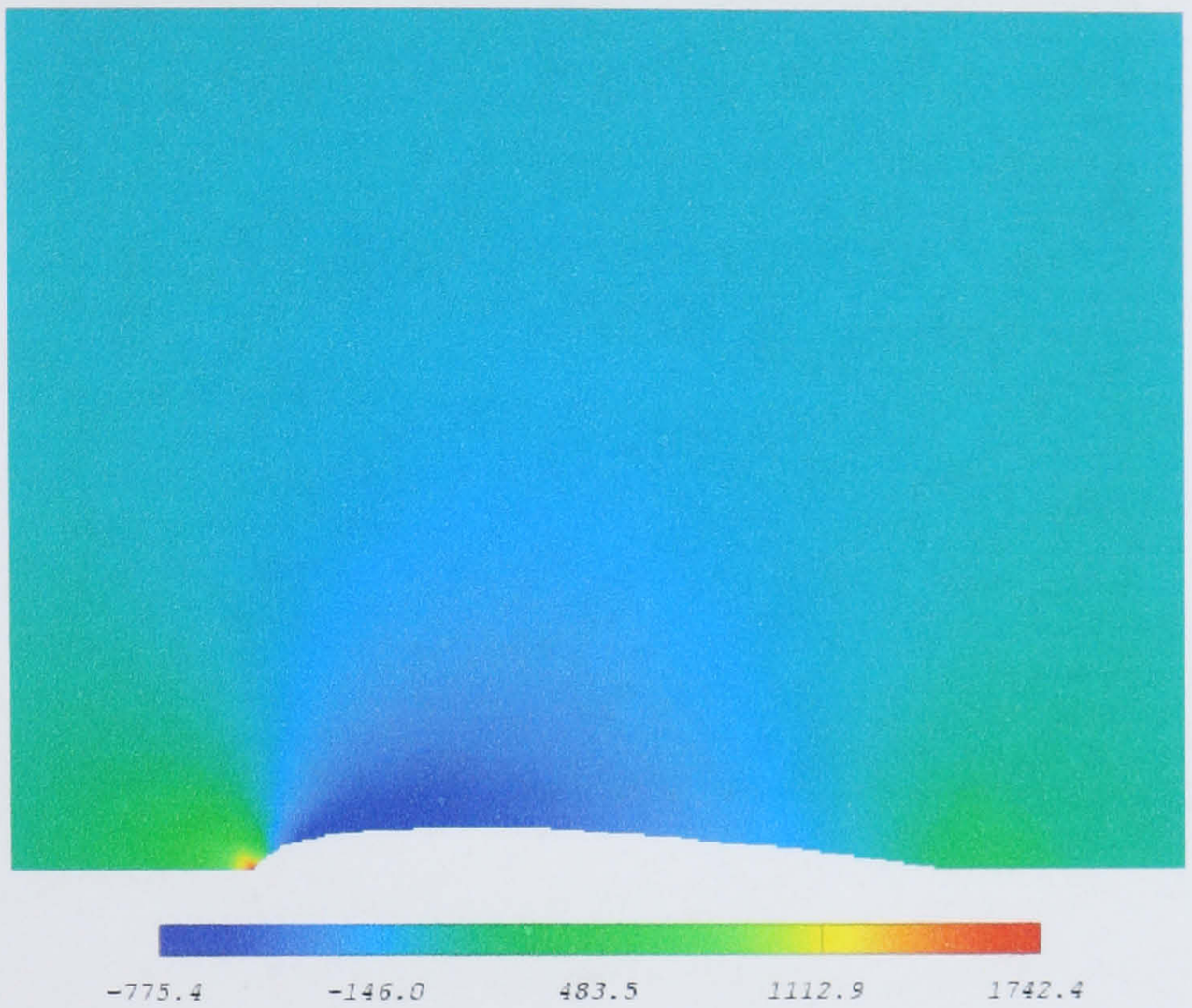
(a)

(b)

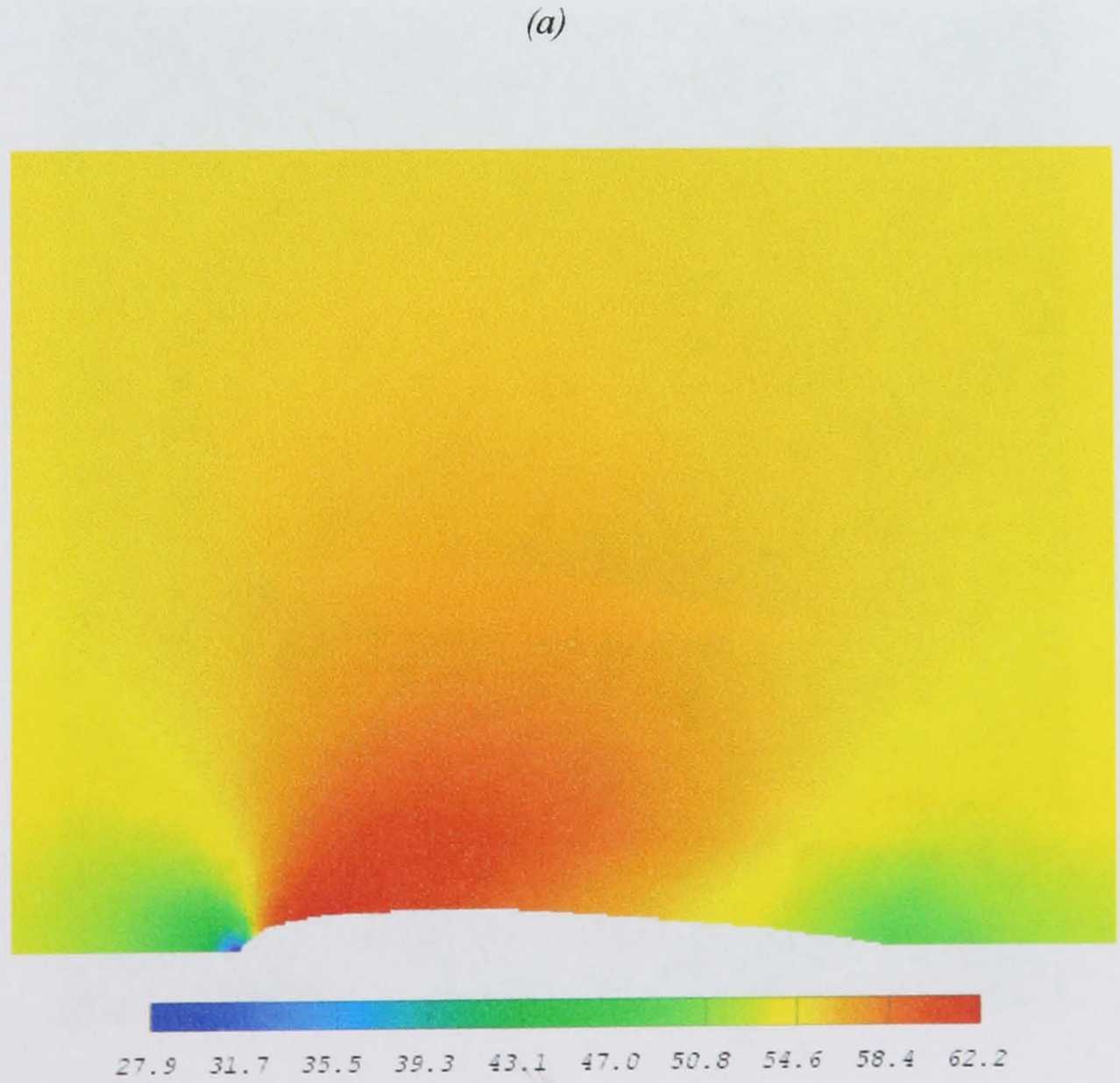
Figure 6.3: The Unstructured Grid Around the 6° Incidence Case (a) Close to the Airfoil (b) In the Complete Computational Domain

Three grids were generated around the airfoil for both the zero degrees and the six degrees incidence cases. The total number of control volumes used in the six grids was 3128, 6334 and 9429 for the zero degrees incidence case and 6854, 15754 and 28962 for the six degrees incidence case. Examples of the coarsest meshes for both cases are shown in Figure 6.2 and Figure 6.3. Each figure shows a close up of the grid around the airfoil and a view of the whole mesh. Clearly the grid has been

refined around the airfoil in comparison to the edges of the domain but more particularly mesh has been concentrated at the leading edge of the airfoil in order to resolve the rapidly changing fluid properties in this region.

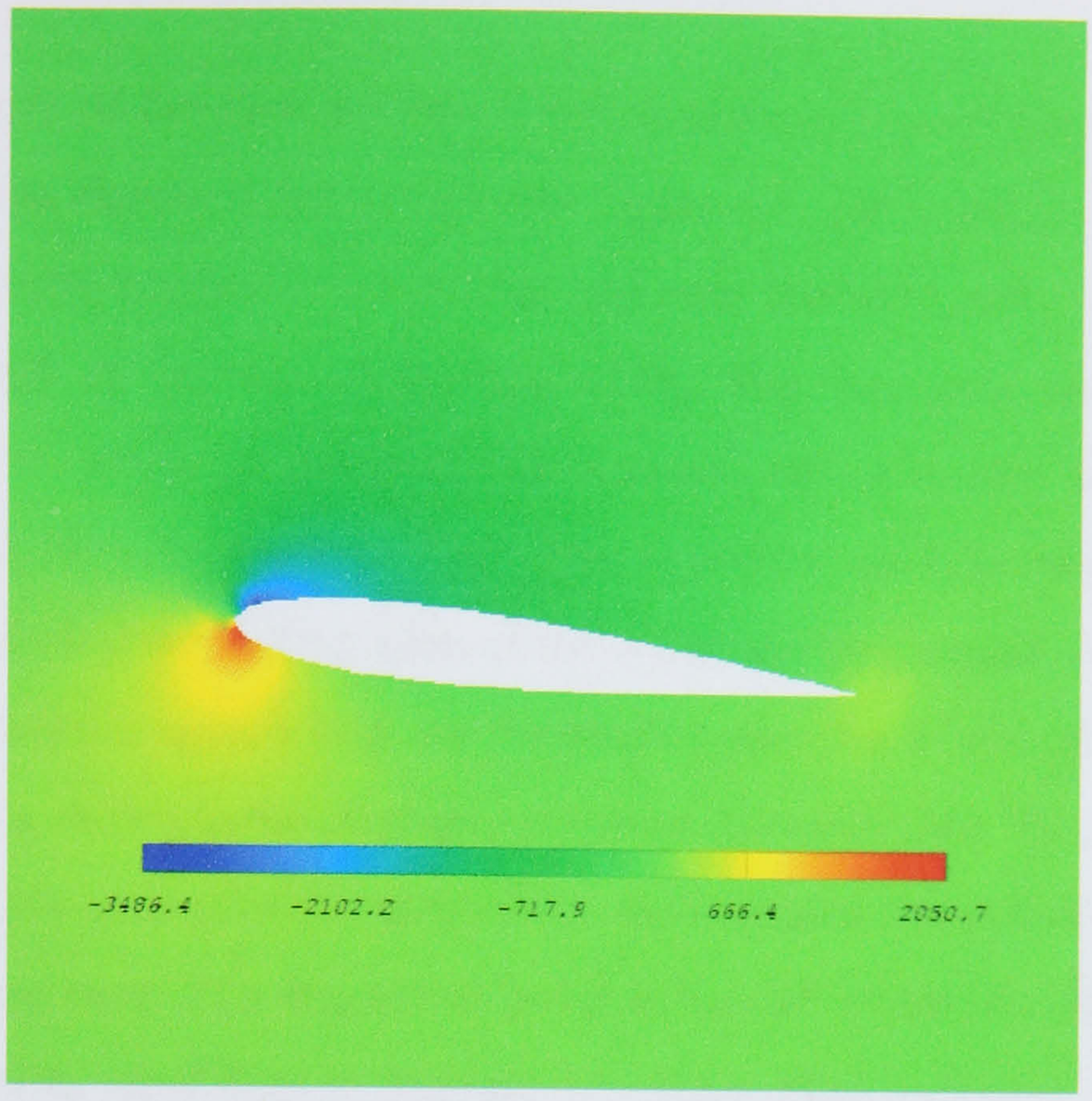


(a)

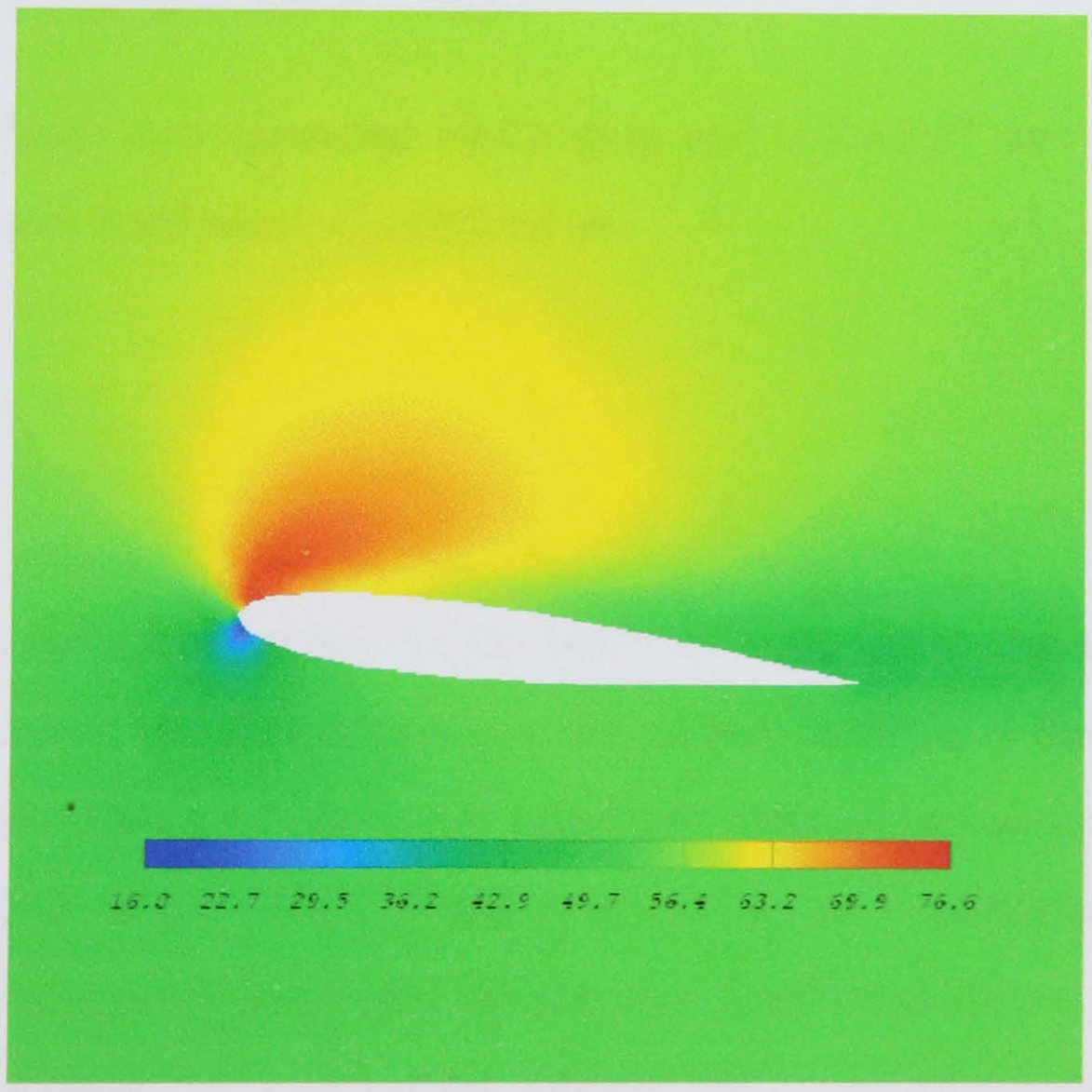


(b)

Figure 6.4: Contour Plots over the 0° Incidence Case (a) Relative Pressure (Pa) (b) Speed (m/s)



(a)



(b)

Figure 6.5: Contour Plots over the 6° Incidence Case (a) Relative Pressure (Pa) (b) Speed (m/s)

The gross details of the flow patterns for both angles of incidence are presented in the contour plots shown in Figure 6.4 and Figure 6.5. These results were calculated using second order convective modelling and used the 6334 and 15754 CV meshes for the zero and six degrees cases respectively. The stagnation point for the zero degrees case, as would be predicted by classical airfoil theory, is located at the leading edge of the wing section. There then follows a rapid decrease in pressure as the fluid moves over the leading edge of the wing, and then a gradual rise in pressure as the trailing edge is approached. As Bernoulli's equation would predict the inverse of this pattern is shown in the corresponding speed contour plot. The results from the six degrees angle of incidence calculations show similar characteristics to those of the zero degrees case. The stagnation point is located below the leading edge and there is then a very rapid drop in pressure over the suction surface of the wing, whilst a more moderate drop in pressure takes place on the pressure surface. The pressure then adjusts itself on both surfaces of the wing to arrive at the trailing edge at an equal value thus satisfying the kutta condition. Again the inverse of the pressure field is shown in the equivalent speed contour.

The static pressure data measured by Gregory and O'Reilly[98] was presented in the form of a pressure coefficient, C_p, defined as

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2} \qquad (6.3)$$

where p is on the upper surface of the blade and p_∞ is the free stream pressure, which in the case of the computations presented here is taken as the specified outlet pressure i.e. zero. Figure 6.6 shows the calculated surface pressure coefficient at zero degrees incidence on the 6334 CV grid compared to the experimental result for all three convective models. In general the pressure coefficient shows good agreement with the experimental data. All models slightly under predict the value of C_p at the leading and trailing edge. This would be expected to improve with grid refinement because the strong gradients present in these regions would be more readily resolved. However a second possible cause is inaccurate turbulence modelling resulting from the use of wall functions.

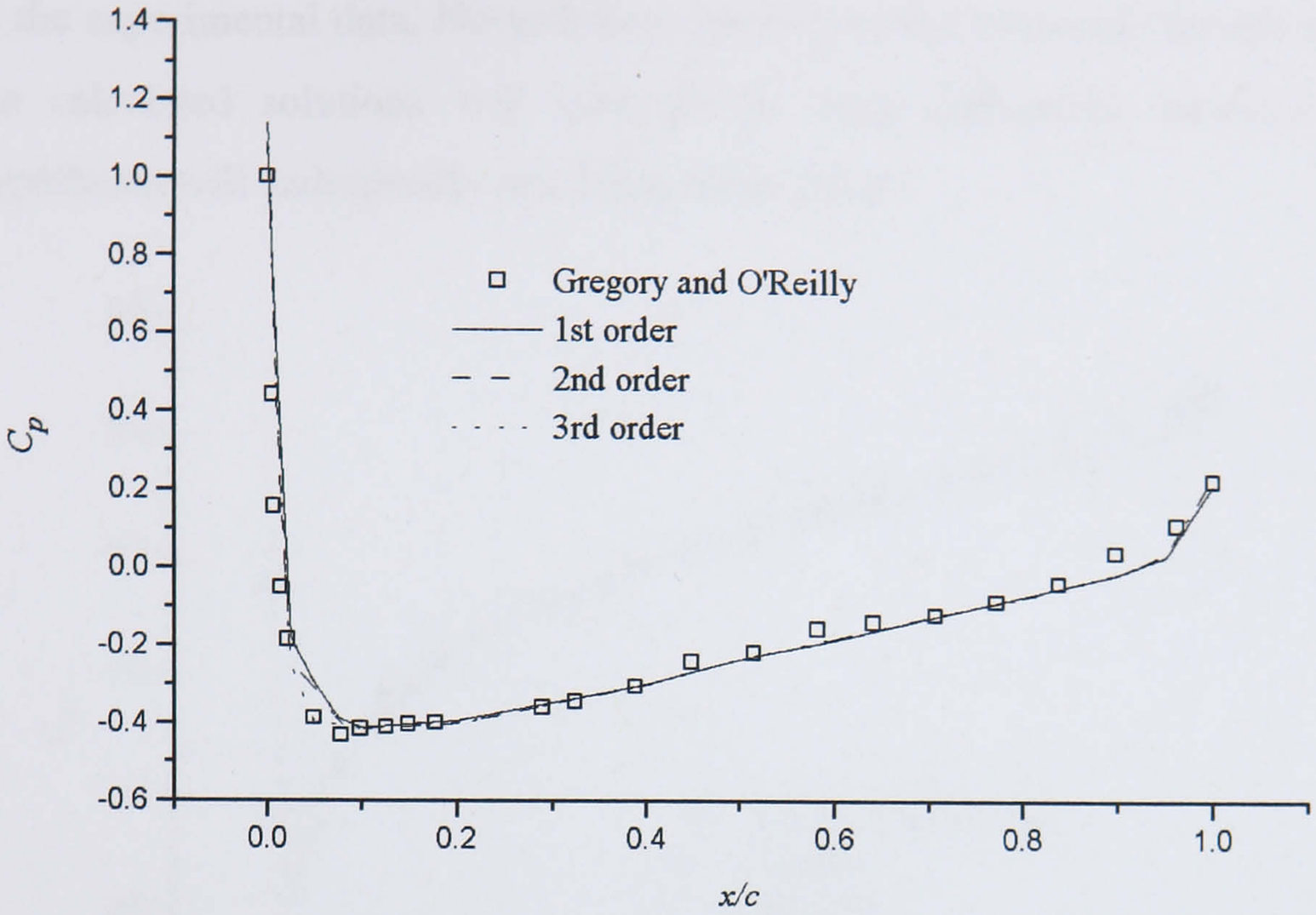


Figure 6.6: Surface Pressure Coefficient at 0° Angle of Incidence

The results of the three convective models shown in Figure 6.6 are virtually identical. This is in stark contrast to the results at high Reynolds number shown in Chapter 5. Four possible reasons are proposed for this difference. Firstly it is only the convective term that has been discretised with different orders of accuracy. All other terms are discretised with constant second order accuracy and therefore these terms may dominate the accuracy of the solution, rather than convective term, leading to comparable results. Secondly, the wall functions used in the control volume immediately adjacent to the airfoil will exert a significant influence over the results presented. Therefore, as all of the convective schemes use the same wall function formulation this will inevitably also produce similar results. Thirdly, the global influence of turbulence modelling on the solution algorithm is to increase the effective viscosity of the flow field, by adding a turbulent viscosity to the actual viscosity. This will in turn clearly lower the effective Reynolds number of the flow, and as was seen in Chapter 5 all convective models produce very similar results at low Reynolds number. Finally, the effects of numerical diffusion are generally only significant in the presence of sharp velocity gradients, and with the exception of the leading edge of the airfoil there are no large velocity gradients in the calculated solution. Close inspection of the leading edge pressure coefficients reveals that in fact the first order model over predicts C_p whilst the higher order models are closer

to the experimental data. Nevertheless the first reason presented for the similarity of the calculated solutions will certainly be most influential, however the other hypotheses will undoubtedly also have some affect.

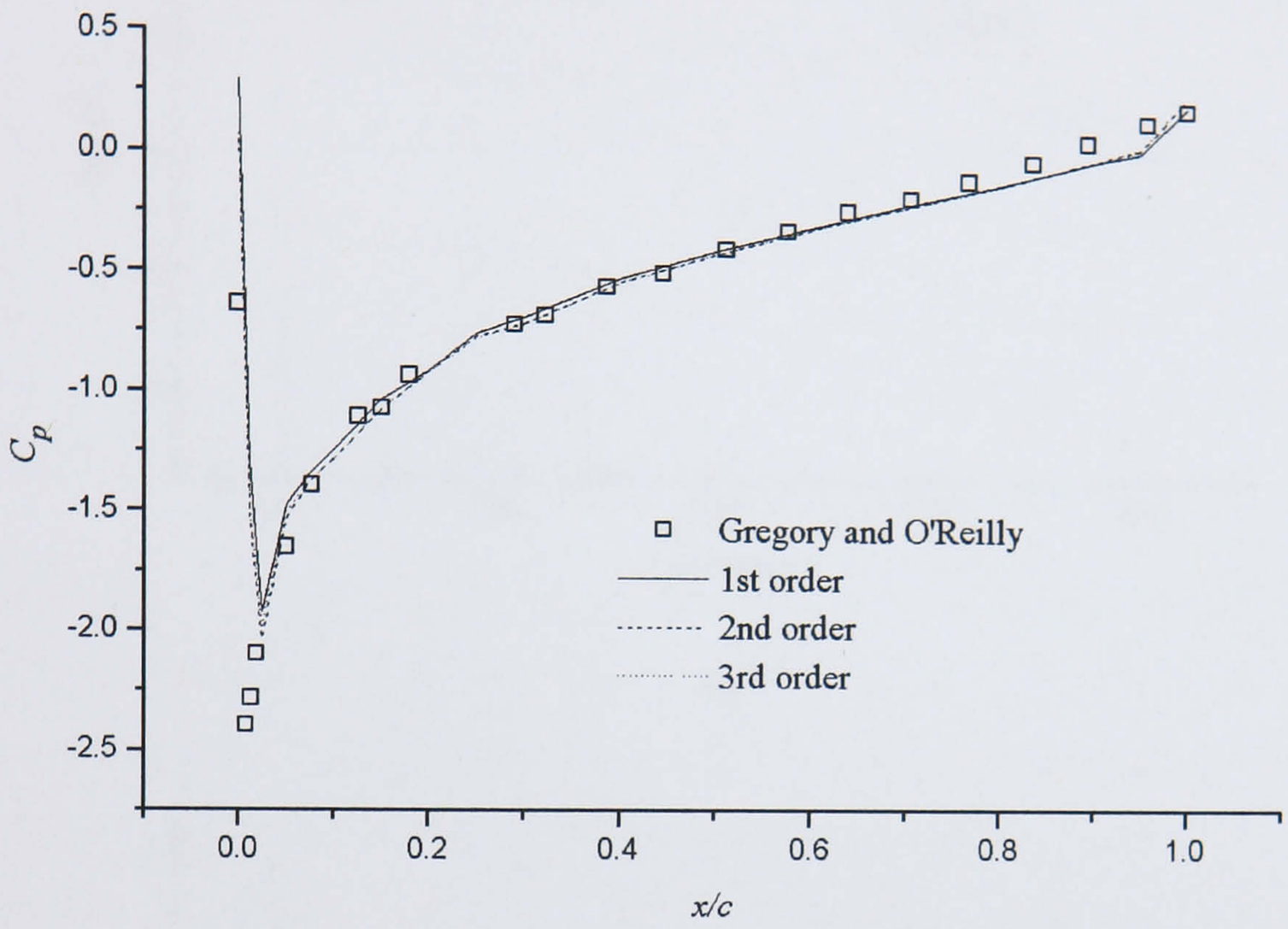
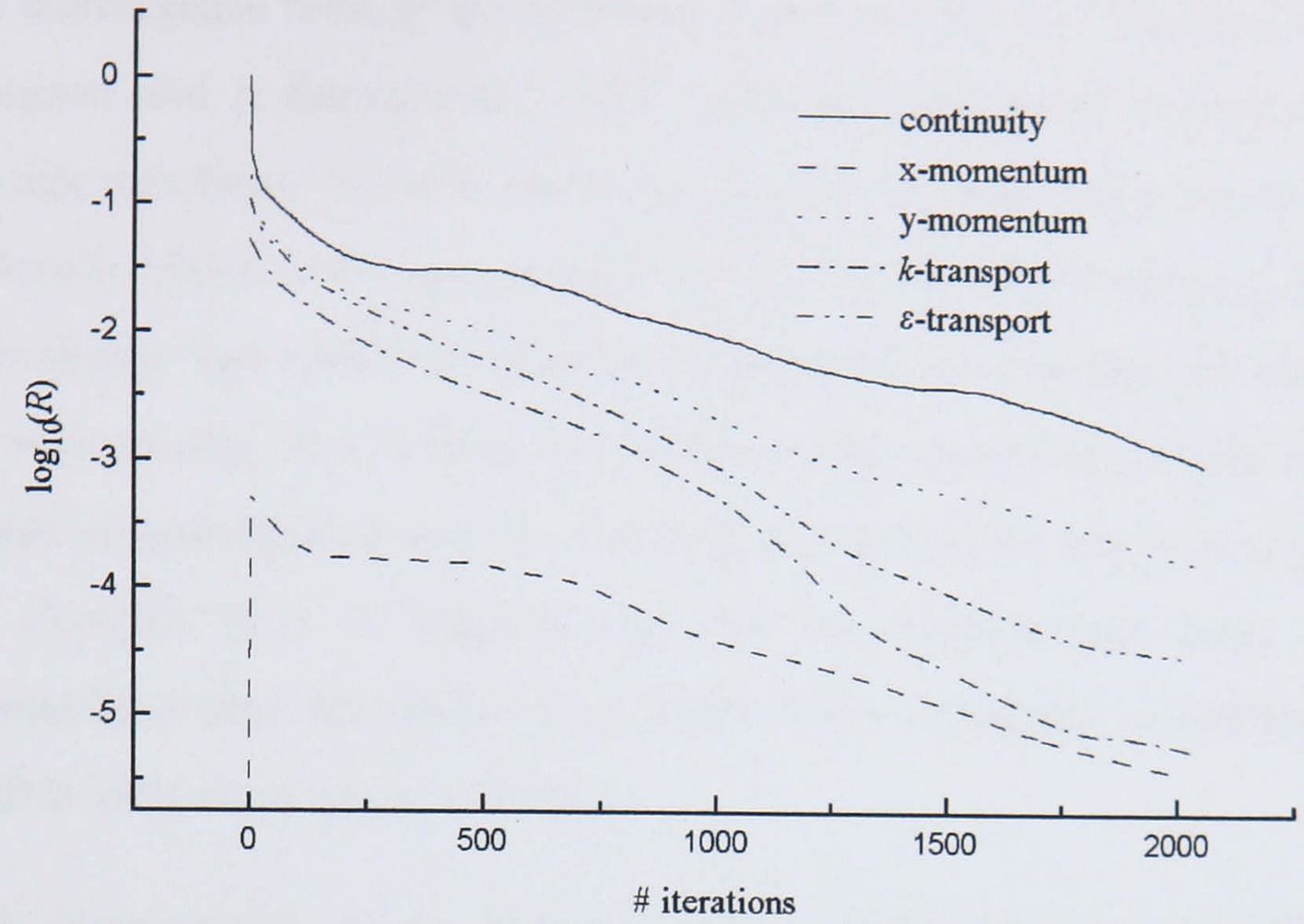
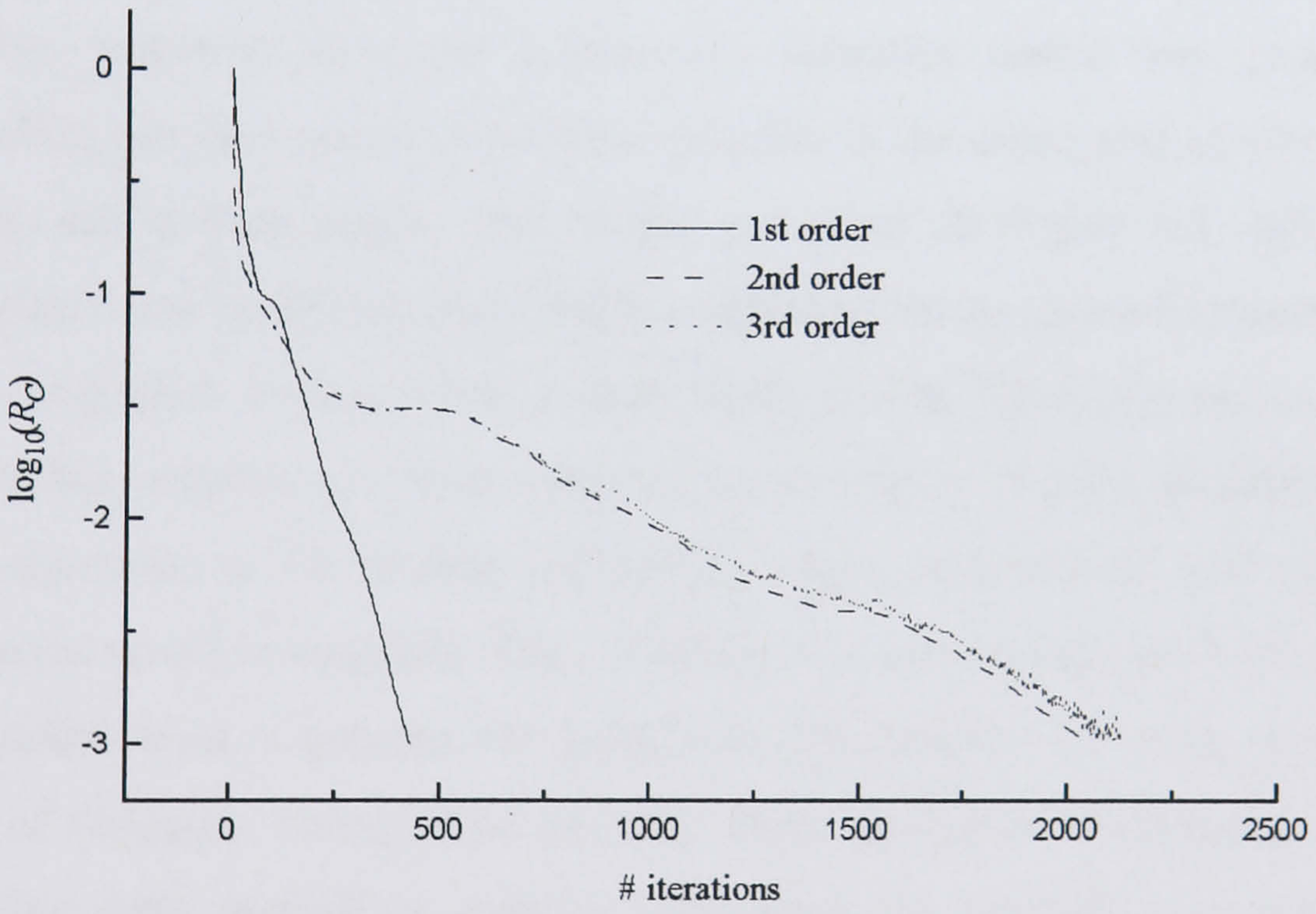


Figure 6.7: Surface Pressure Coefficient at 6° Angle of Incidence

Figure 6.7 shows an equivalent comparison of the three convective models to Figure 6.6, for the six degrees incidence case. These results were calculated using the 15754 CV grid and in general good agreement is shown between Gregory and O'Reilly's[98] data and the calculated pressure coefficients. However, all the convective models over predict the pressure coefficient around the leading edge and slightly under predict it at the trailing edge. These discrepancies are in areas where there are rapid changes in the flow field and clearly the solution algorithm has difficulty in resolving these changes with high accuracy. This was also the situation for the zero degree case although the inaccuracies were less pronounced. Nevertheless, the potential causes of the inaccuracies are the same in both cases i.e. inaccuracies in the near wall turbulence model and inadequate grid refinement in this region. Similarly, the three convective models in the six degree case produce virtually identical results as they did in the zero degrees case. The possible causes of this effect have already been highlighted for the zero degree case and therefore will not be repeated.



(a)



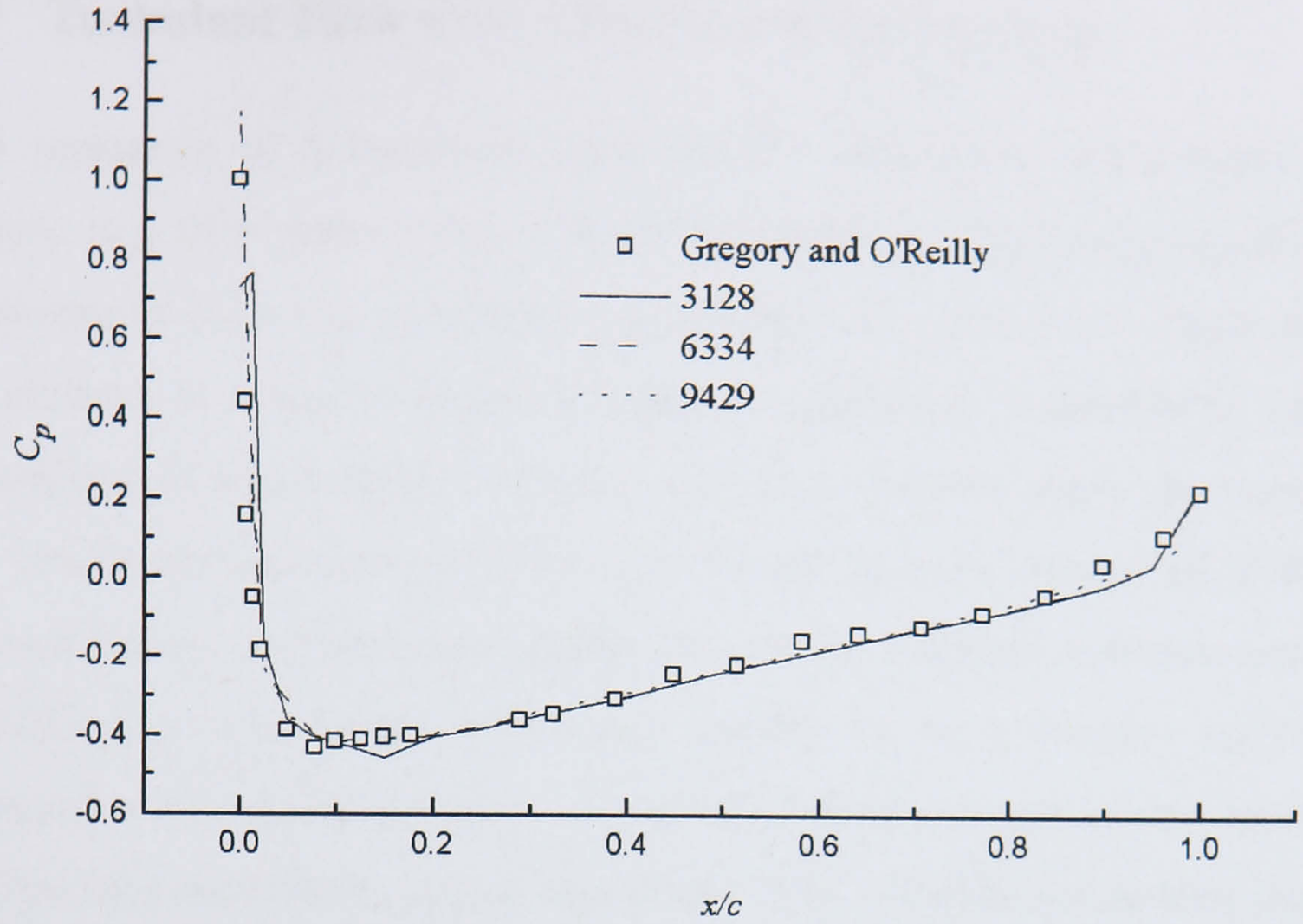
(b)

Figure 6.8: NACA 0012 Convergence Histories (a) Comparison between Governing Equations (b) Comparison between Convective models

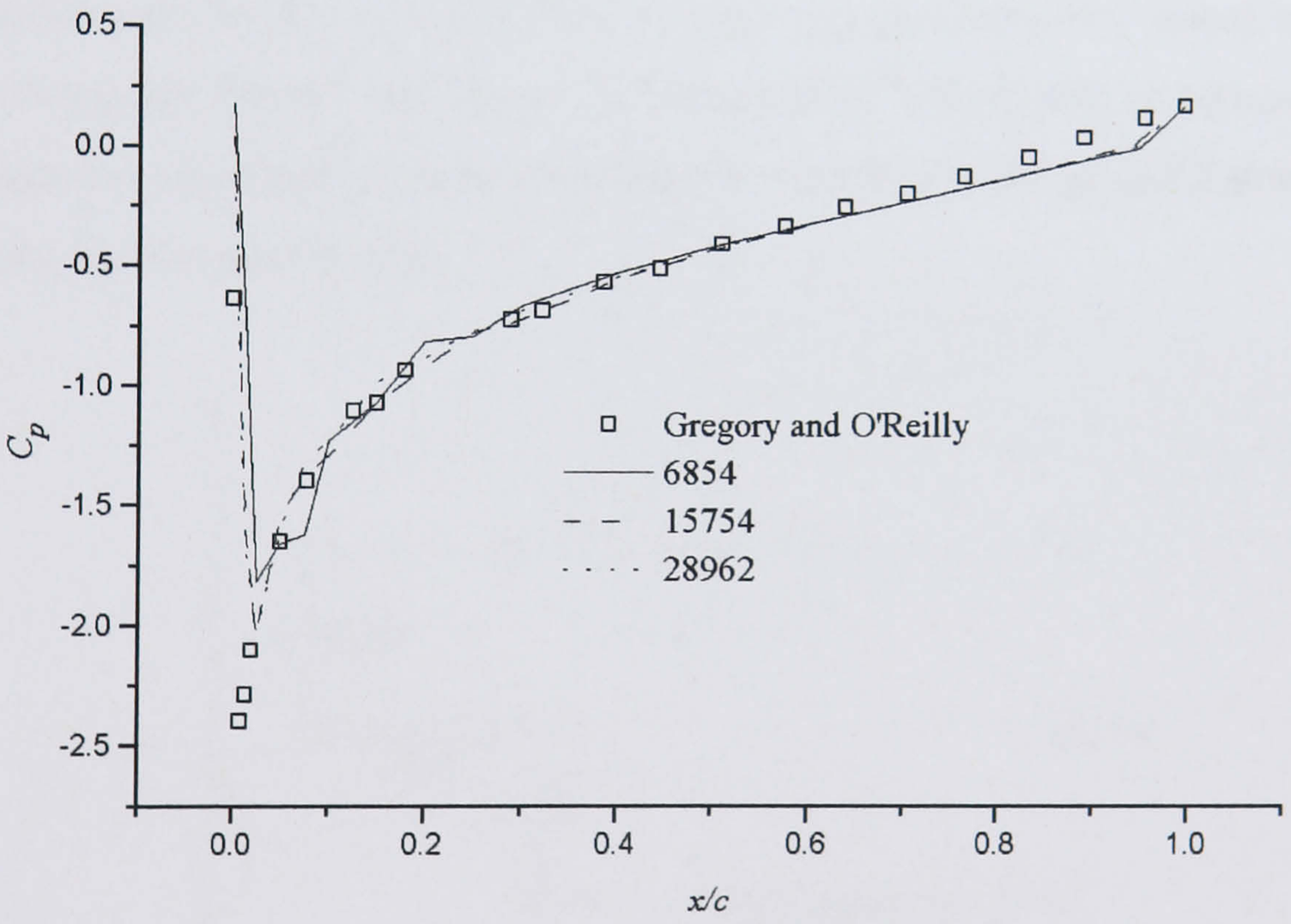
Figure 6.8a shows the convergence history of all six governing equations for the zero degrees incidence case using the 6334 CV grid and second order convective modelling. As was the case in laminar flow the continuity equation has the slowest convergence rate of the six governing equations and will be used here for comparison

of the convergence rates of the different convective models. Figure 6.8b shows this comparison and it demonstrates once again that the more explicit nature of the governing equations, brought about by the use of high order convective models, significantly effects convergence rates. In this test case the continuity equation takes approximately four times the number of iterations to converge for the second and third order models than it does for the first order model. In laminar test cases this decrease in convergence rate for the high order models was normally a factor of three. However it is no surprise that this has become four here, as two extra equations have been introduced as a result of the turbulence modelling, which will inevitably increase convergence rates.

Finally, Figure 6.9 shows a comparison of the calculated pressure coefficients, using second order convective modelling, for the three grids used in both the zero and six degrees test cases. Clearly, this shows that in both cases a grid independent solution has been achieved. However, it has been identified earlier that inadequate grid refinement maybe a cause of the discrepancies in the calculated results around the leading and trailing edges. The results presented in Figure 6.9 still show this discrepancy and would therefore tend to point towards the grid refinement hypothesis being misguided. In fact, what is more likely is that Figure 6.9 shows that a grid independent solution has been achieved for this form of grid. In order to achieve better resolution at the leading and trailing edges, concentrated grid refinement in these areas would be required. This could not be achieved with the form of Delaunay triangulation used to generate the grids here. Nevertheless by using more advanced forms of Delaunay triangulation or other mesh generation techniques, such as the advancing front method or adaptive refinement, the required resolution could be achieved. However the development of an appropriate grid generation tool to do this is beyond the scope of this thesis but it would be an interesting area of further research.



(a)



(b)

Figure 6.9: Surface Pressure Coefficient as a Function of Grid Fineness (a) 0° Angle of Incidence (b) 6° Angle of Incidence

6.3 Turbulent Flow over a Backward Facing Step

The separation of a boundary layer and it's subsequent reattachment to a solid surface is a flow pattern that is found in numerous engineering applications. The phenomenon occurs in propellers, gas turbines and pipelines to name just a few of the plethora of examples found throughout engineering. Separation in wing sections causes loss of aerodynamic efficiency and increased drag whilst in internal flows it can causes heat transfer problems and thereby reduces the overall efficiency of a thermodynamic system. Consequently, the ability to predict turbulent separated flow in addition to it's laminar counterpart, is essential for a solution algorithm that is designed to be applied to a wide variety of problems, as any unstructured algorithm is. The backward facing step is one of the least complex geometries that results in separated flow and has become a standard test case for turbulent as well as laminar solution algorithms. A number of authors have published experimental measurements for the turbulent flow through this geometry two examples of which are Vogel and Eaton[99] and Driver and Seegmiller[100]. Both sets of authors produced comparable experimental results and therefore arbitrarily Vogel and Eaton's[99] data is chosen for comparison here.

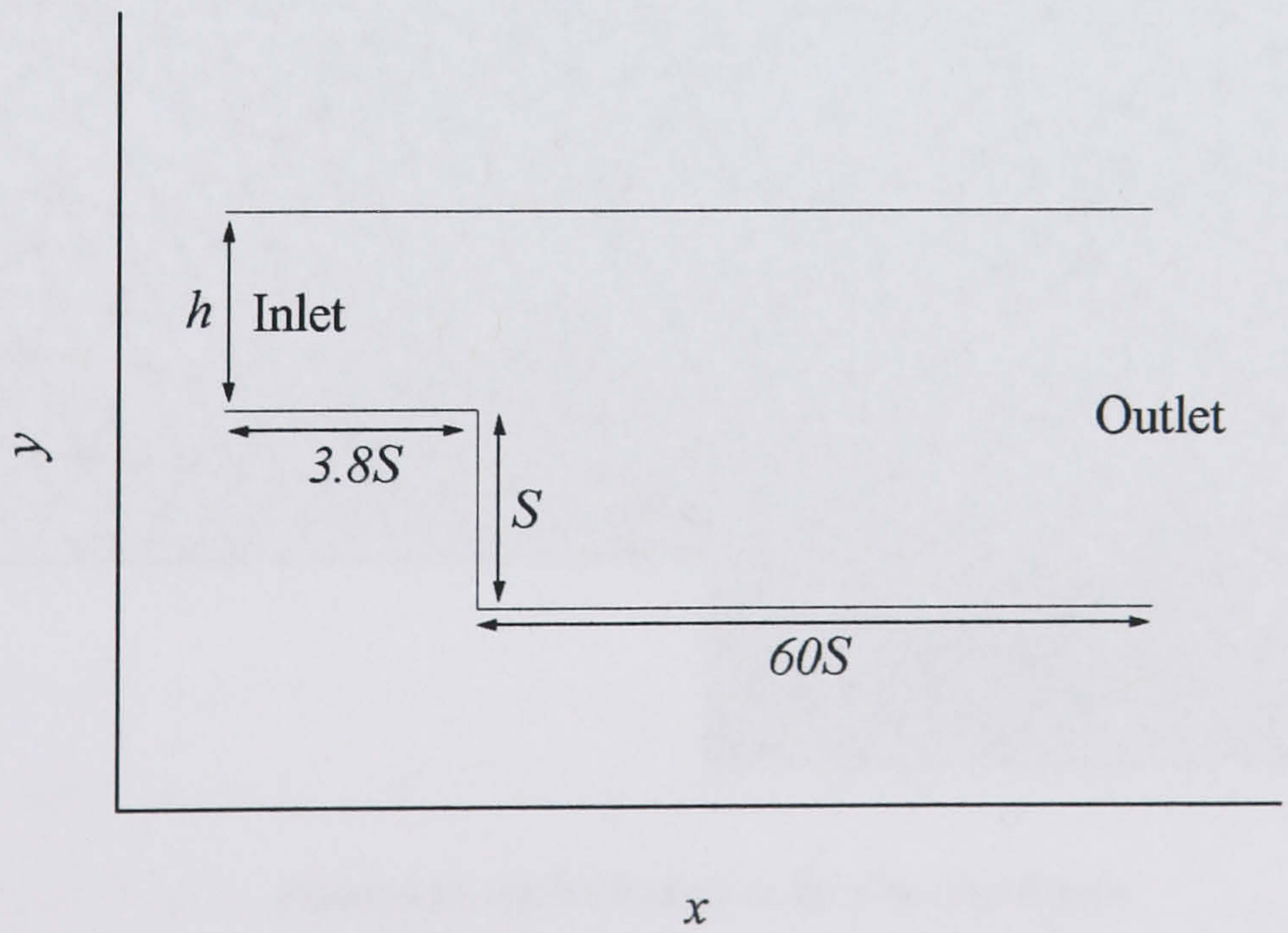


Figure 6.10: Schematic of the Turbulent Backward Facing Step

A schematic of the geometry used in this test case is shown Figure 6.10. It differs from the laminar case in that the outlet section has been increased in length to sixty

step heights and an inlet section has been added of 3.8 step heights. The increase in outlet section is to allow sufficient distance after the separation region so that a fully developed flow profile will have established itself before the outlet boundary condition is applied. Similarly, the inlet section has been added to allow a fully developed flow profile to develop in the channel downstream of the inlet boundary condition. This inlet section was not required in the laminar test case because a fully developed Poiseuille flow profile could be specified at the step. The applied boundary conditions were $u=U_\infty$, the free stream velocity outside the boundary layer, and $v=0$ at inlet, constant pressure at outlet and no-slip condition with standard wall functions along all the walls. The Reynolds number of the calculation was set at 28,000 based upon the free stream velocity outside the boundary layer and the step height, S. The expansion ratio of the test case was 1:1.25 which results in $h=3.95$ if $S=1$. Each calculation performed was iterated until the residual of each of the governing equations was reduced to 10^{-3}.

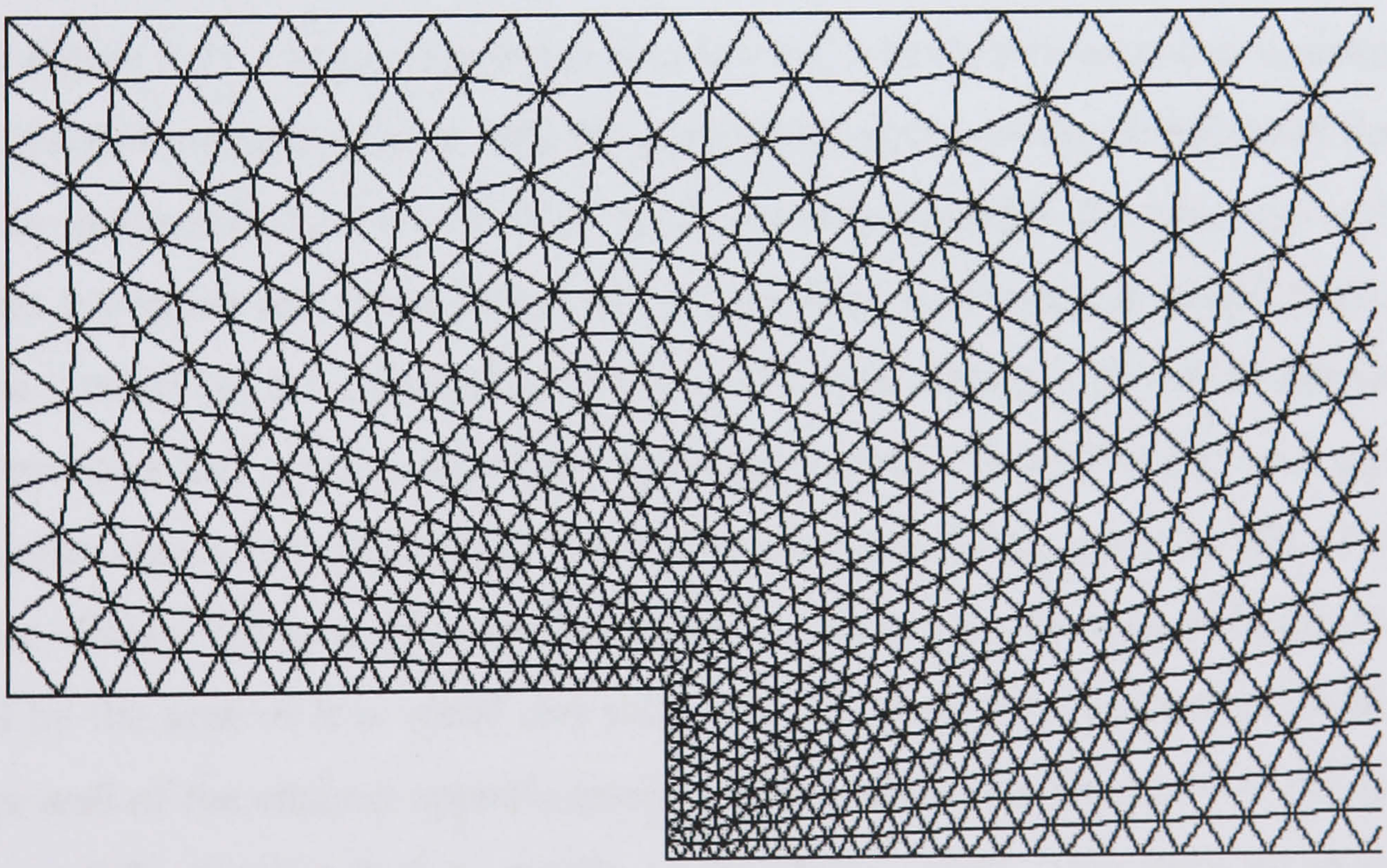
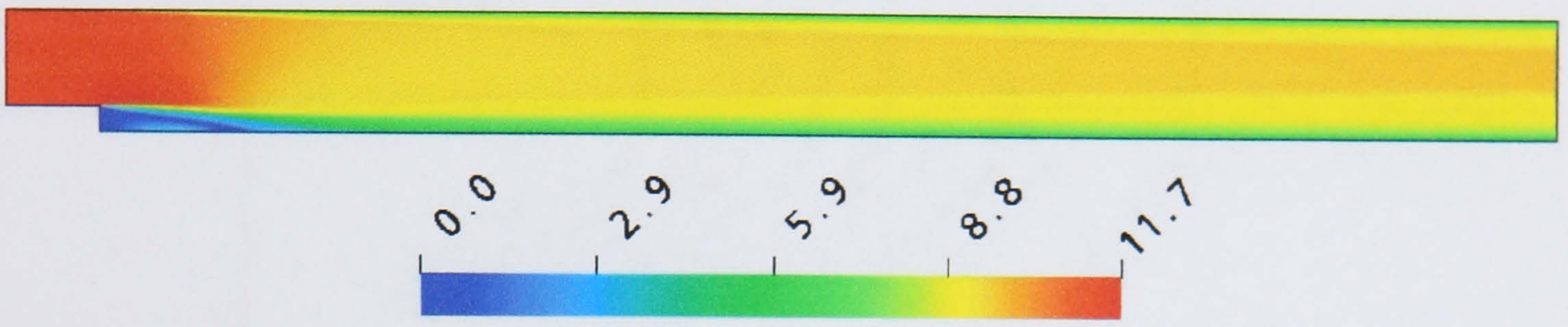


Figure 6.11: 5513 CV Mesh in the Near Step Region

(a)



(b)

Figure 6.12: Contours over the Backward Facing Step (a) Relative Pressure (Pa) (b) Speed (m/s)

Three grids were used to calculate solutions to this test case, they contained 5513 CV, 10135 CV and 17563 CV. The mesh around the step area in the 5513 CV grid is shown in Figure 6.11. Clearly this shows that fine mesh has been concentrated around the step and downstream of it, so that the greatest resolution of the complicated flow patterns in this region can be achieved. As the mesh approaches the inlet and outlet boundaries it become less dense reflecting how the flow becomes less complicated as these two boundaries are approached. Figure 6.12a and Figure 6.12b show contour plots of pressure and speed respectively calculated on the 10135 CV grid and using second order convective modelling. These two plots show the gross details of the flow quite clearly. An area of separation develops behind the backward facing step indicated by the area of low speed and pressure. This separation reattaches itself to the lower wall of the channel approximately ten step heights downstream of the step and then a fully developed flow profile re-establishes itself. This fully developed profile is characterised by the constant linear pressure gradient and the distinctive steady speed in the centre of the channel and the rapid reduction to zero in the near wall region.

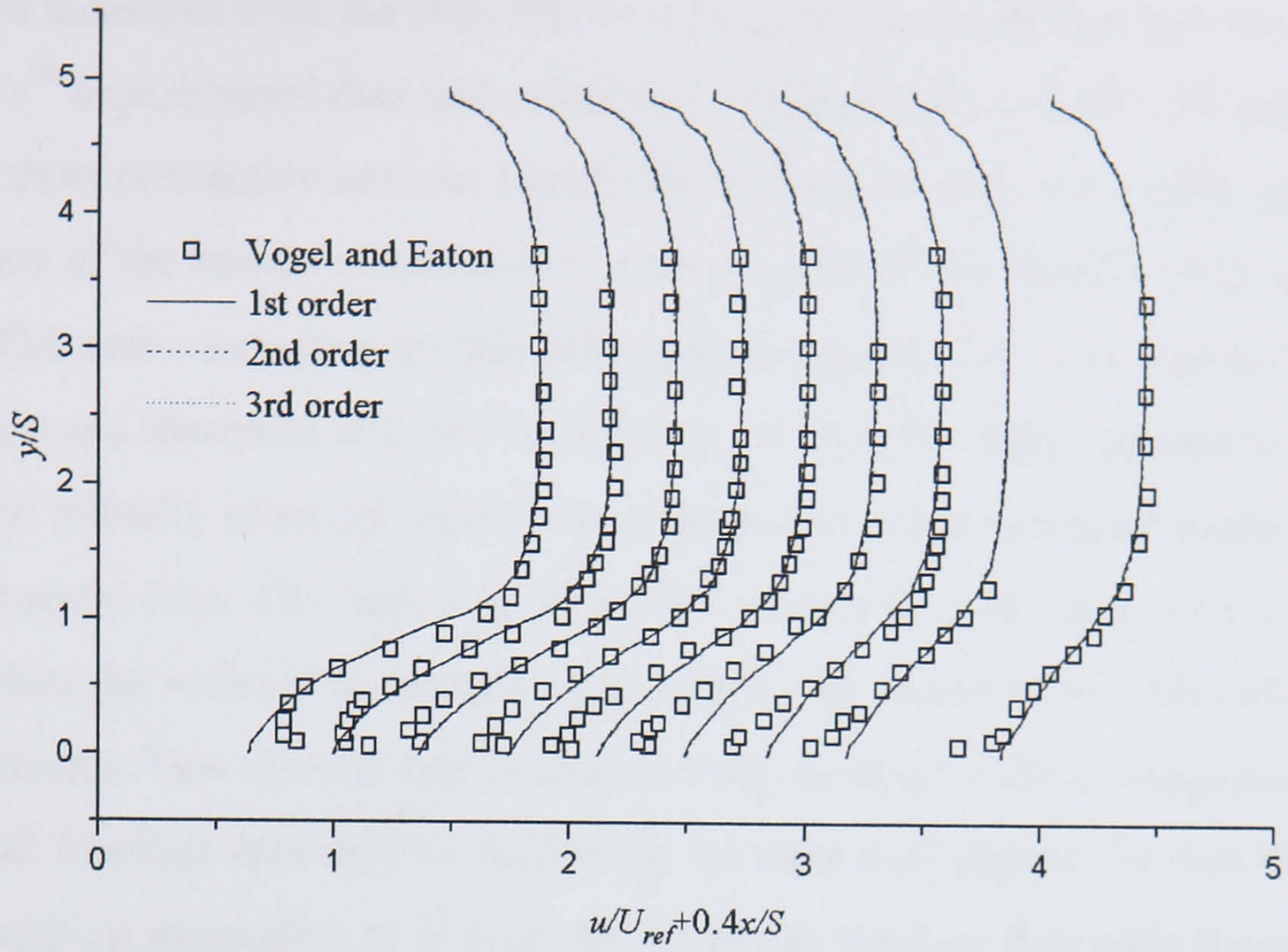


Figure 6.13: Velocity Profiles at Various x/S Locations

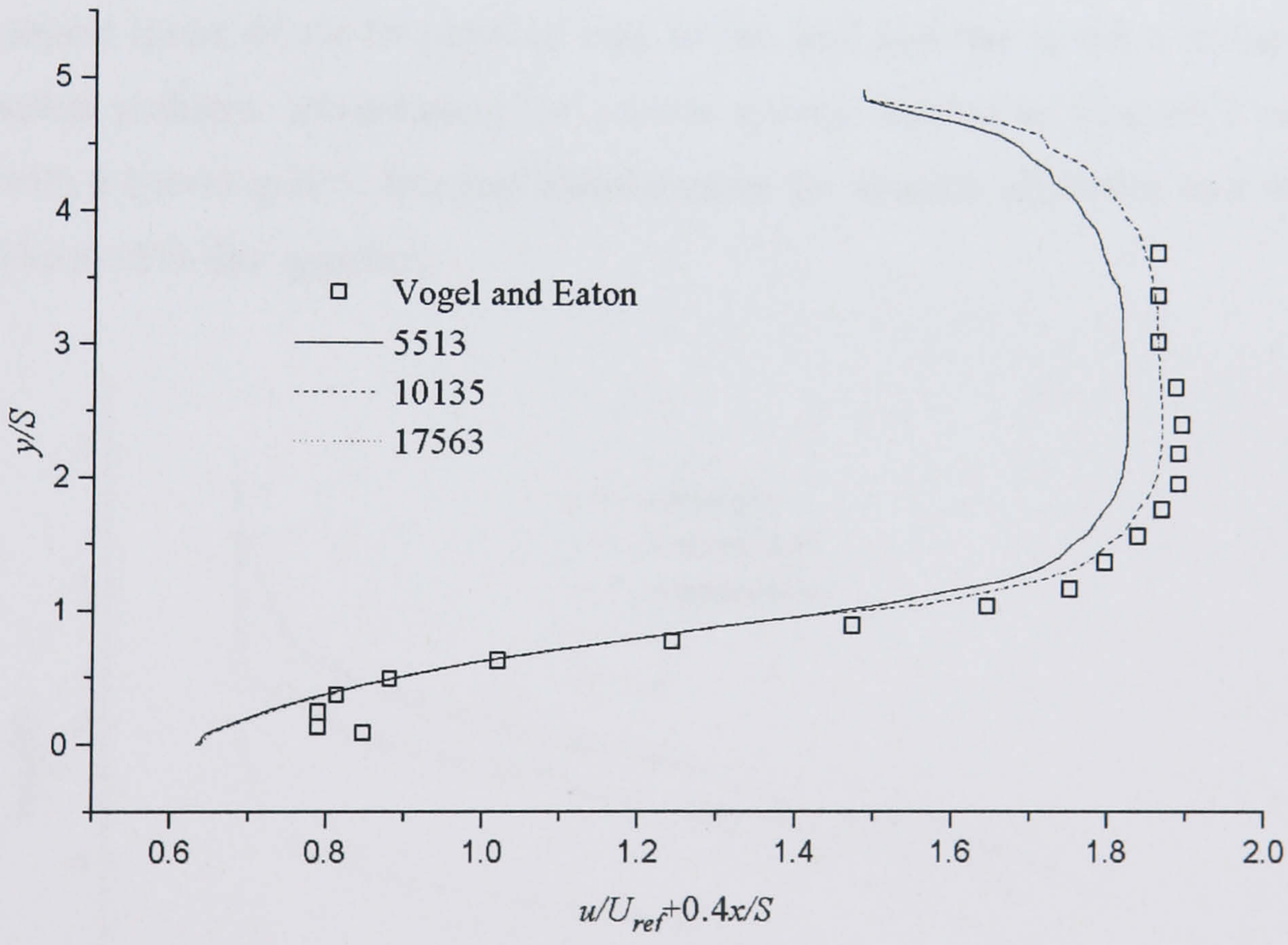
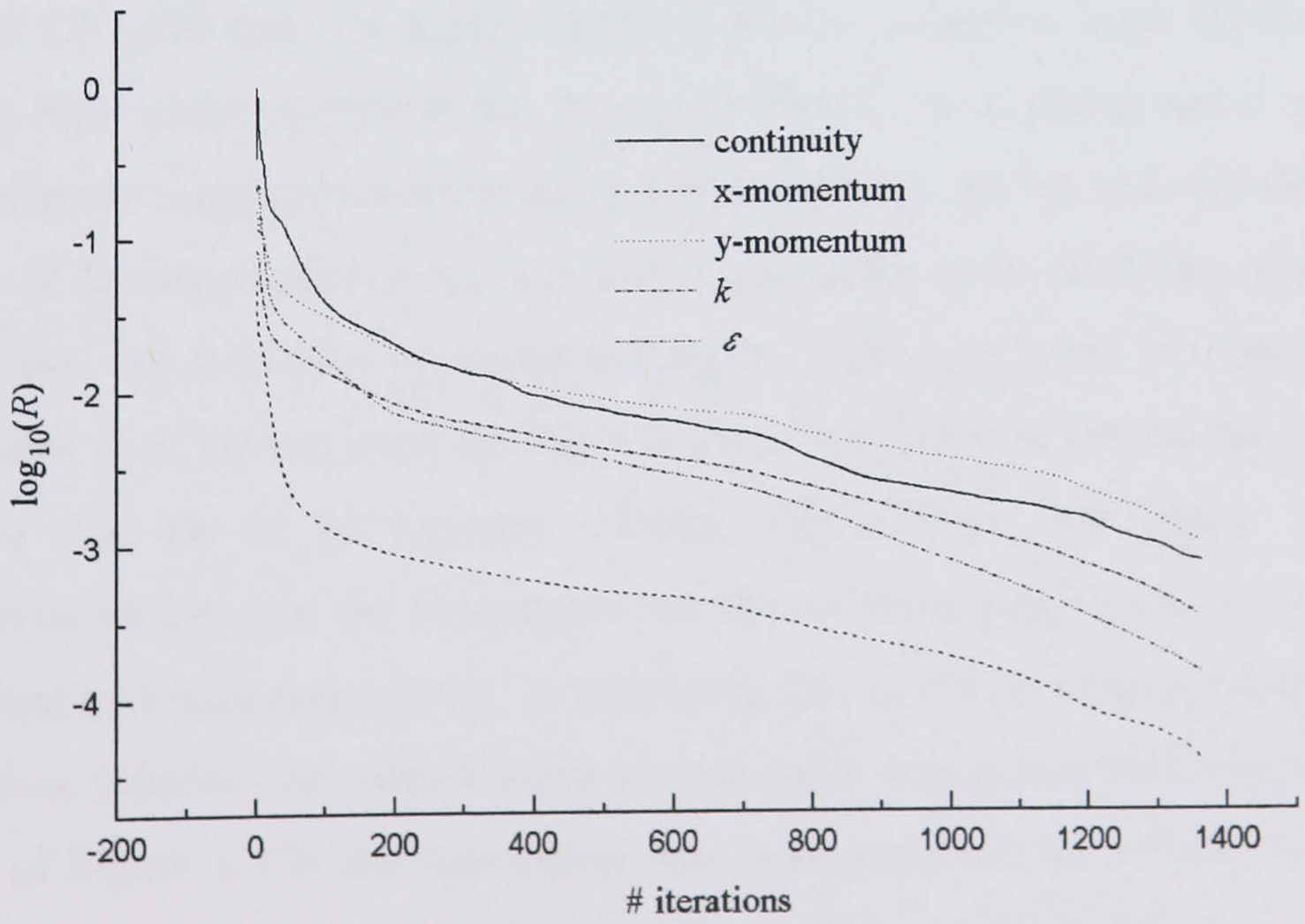


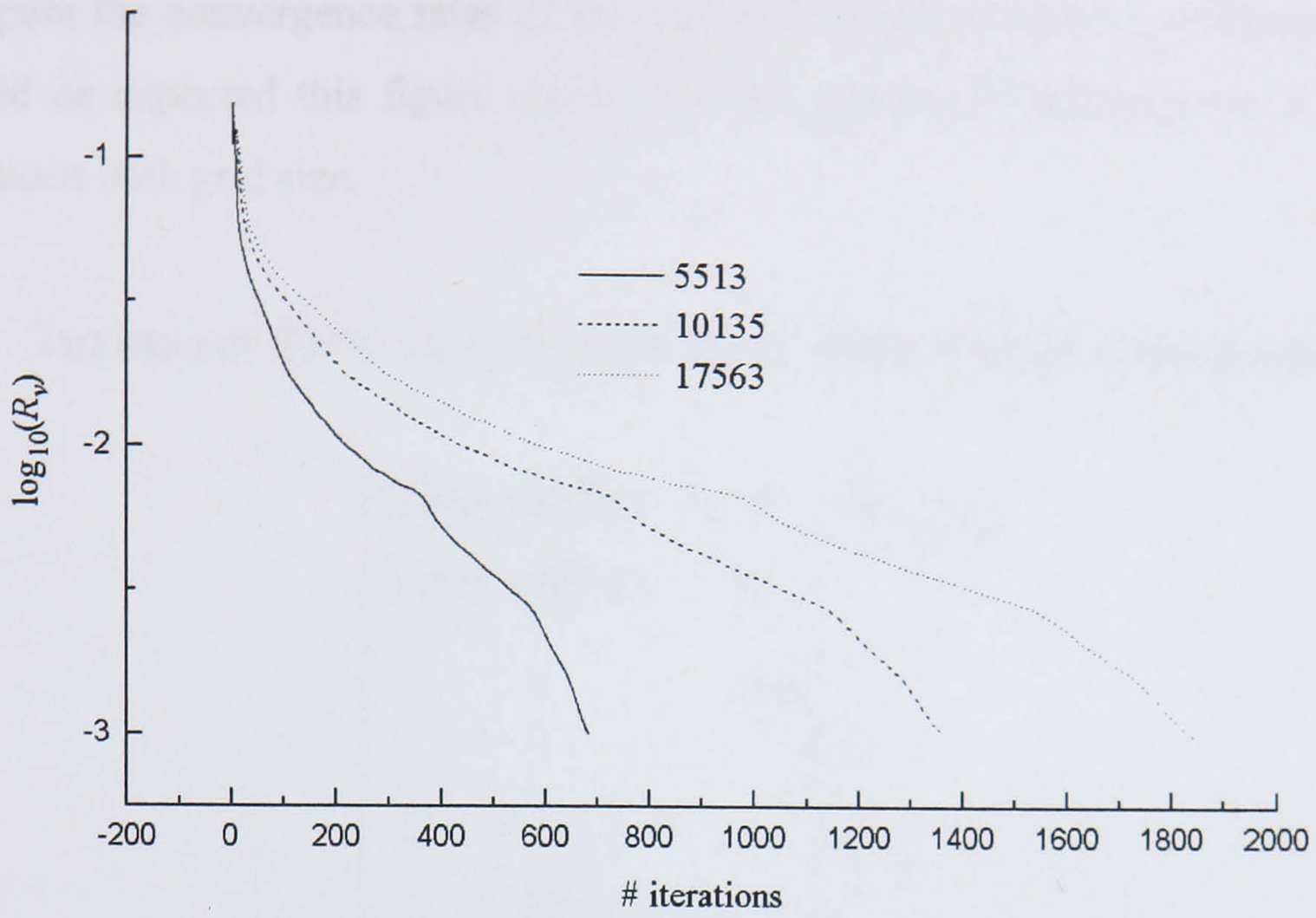
Figure 6.14: Variation of the x/S=2.2 Velocity Profile as a Function of Grid Fineness

Vogel and Eaton[99] measured the velocity profile using laser Doppler anemometry at nine locations along the channel downstream of the step and it is these results that will be used for comparison purposes here. The nine locations were x/S=2.2, 3, 3.73, 4.46, 5.19, 5.93, 6.66, 7.39 and 8.86 where x is the horizontal distance along the

channel measured from the step. Figure 6.13 shows a comparison between Vogel and Eaton's[99] experimental data and calculated results using the 10135 CV mesh for each of the three convective models. Clearly the influence of the convective model on the accuracy of the results is minimal as was the case in the NACA 0012 aerofoil test case. The same reasoning for this effect can be applied here as it was in the aerofoil test case and therefore will not be repeated. Whilst the three convective models all produce virtually identical results, by in large this result is a good prediction of the experimental data. The region in which the models all perform poorly is near to the wall. Here the velocity is generally larger than was measured in Vogel and Eaton's[99] experiments. This discrepancy is undoubtedly caused by the inadequacies of using the wall function approach to modelling the near wall region. As was discussed in Chapter 2 an alternative to this would be to use the low Reynolds form of the k-ε model near to the wall coupled with a suitably refined grid. This approach is not without it's difficulties though, as triangles or more probably quadrilaterals with very high aspect ratios would be required near to the wall and this is not a trivial grid generation problem. Nevertheless the pointer system detailed in Chapter 3 would deal with a hybrid grid of this sort and therefore the solution algorithm as a whole could be used in this approach.



(a)



(b)

Figure 6.15: Backward Facing Step Convergence Histories (a) Comparison between Governing Equations (b) Comparison between different Mesh Sizes

Figure 6.14 shows a comparison of the velocity profile calculated using second order convective modelling at x/S=2.2 for the three different grid sizes used in this test case. Obviously, a grid independent solution has been achieved as the 10135 CV and 17563 CV grid arrive at almost identical results. Solutions from all the grids still suffer from under predicting the velocity in the near wall region and it appears that this effect is completely unaffected by grid refinement. As has already been noted the cause of this inaccuracy in the near wall region is the use of wall functions. It can be seen from the derivation presented in Chapter 3 that grid refinement has little if any influence over the accuracy of wall functions and therefore the results presented in Figure 6.14 are to be expected. Finally, Figure 6.15a and Figure 6.15b show comparisons between the convergence of the six governing equations and the three different grid sizes respectively. In both these figures the convergence rates presented are from solutions calculated using second order convective modelling and in the case of Figure 6.15a the calculation was performed on the 10135 CV grid. The comparison of the six governing equations shows that in this test case, unlike previous results, the y momentum equation takes the largest number of iterations to converge. As a consequence it is this equation that is used in Figure 6.15b to

compare the convergence rates of the calculations performed on different grids. As would be expected this figure shows that the number of iterations to convergence increases with grid size.

6.4 Turbulent Flow in a Square Duct with Strong Curvature

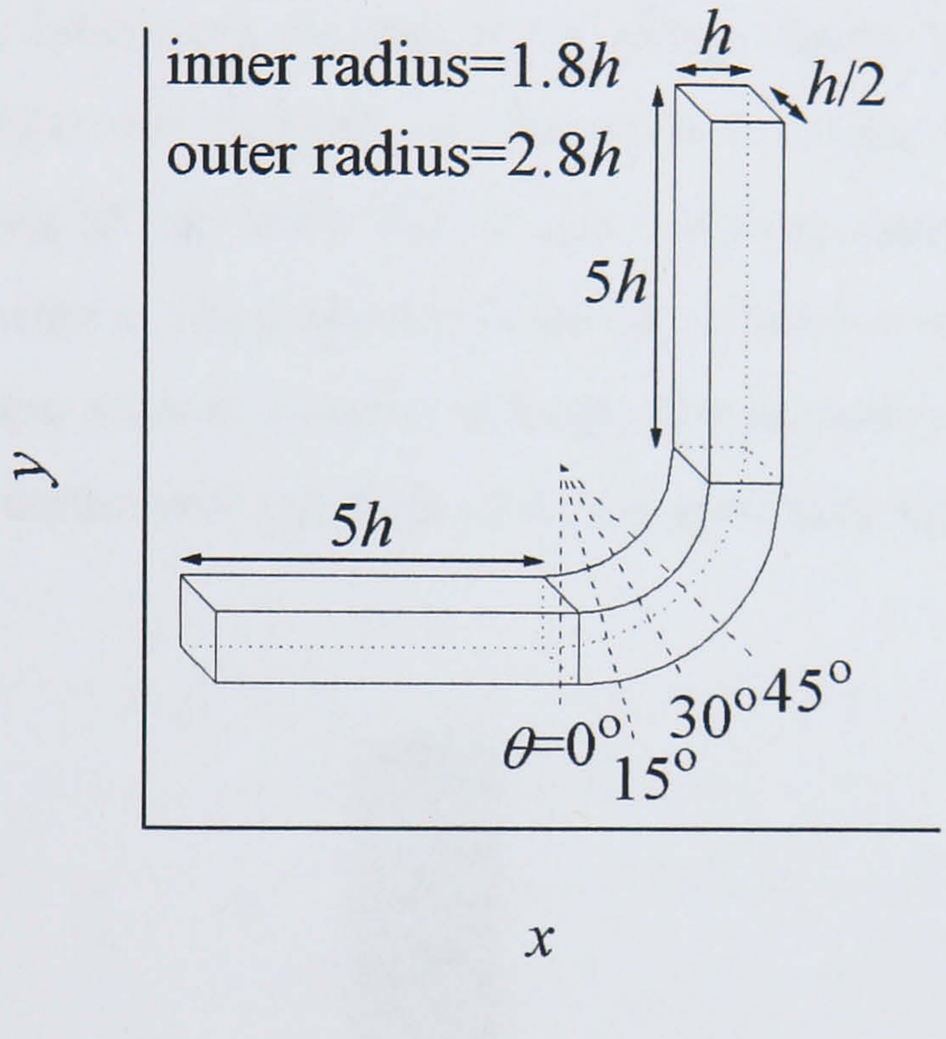


Figure 6.16: Schematic of the Turbulent Square Duct with Strong Curvature

The measurement and prediction of the complex internal flow patterns associated with turbulent flow in a 90° bend has been a challenge for experimental techniques and computational modelling respectively. A number of authors have published experimental data for this configuration including Humphrey *et al.*[101], Taylor *et al.*[102] and Kliafas and Holt[103]. Here, the measurements of Kliafas and Holt[103] will be used for comparison with numerical results. The duct used by these authors for their experimental work had a cross section of 100x100mm^2 and downstream and upstream straight duct sections of 1.2m and 1m respectively. The mean radius of the 90° bend was 92mm i.e. identical to the bend section used in the equivalent laminar test case in Chapter 5. The computational domain used for calculation did not mimic the actual experimental rig in two respects. Firstly the inlet and outlet straight duct sections were reduced to a length of $5h$, where h is the length of one side of the square cross section, and secondly, owing to symmetry only one half of the physical domain was modelled computationally. Both these adjustments are identical to the

changes made to the computational domain in the analogous laminar test case and the schematic of this domain is shown here in Figure 6.16.

Kliafas and Holt's[103] experimental data was measured at a Reynolds number of 347,000 based upon the hydraulic diameter of the duct, $d=h$, and the bulk velocity, U_b. At the inlet of the computational domain a fully developed turbulent profile was specified thereby facilitating the shortening of the straight inlet duct section, as no development section was required. On the outlet boundary a constant pressure was assumed and along all the walls the no slip boundary condition was applied with standard wall functions. The symmetry boundary condition was enforced by ensuring that there was zero normal velocity through this boundary. All calculations were iterated until the normalised residuals of all the governing equations were reduced to 10^{-3}.

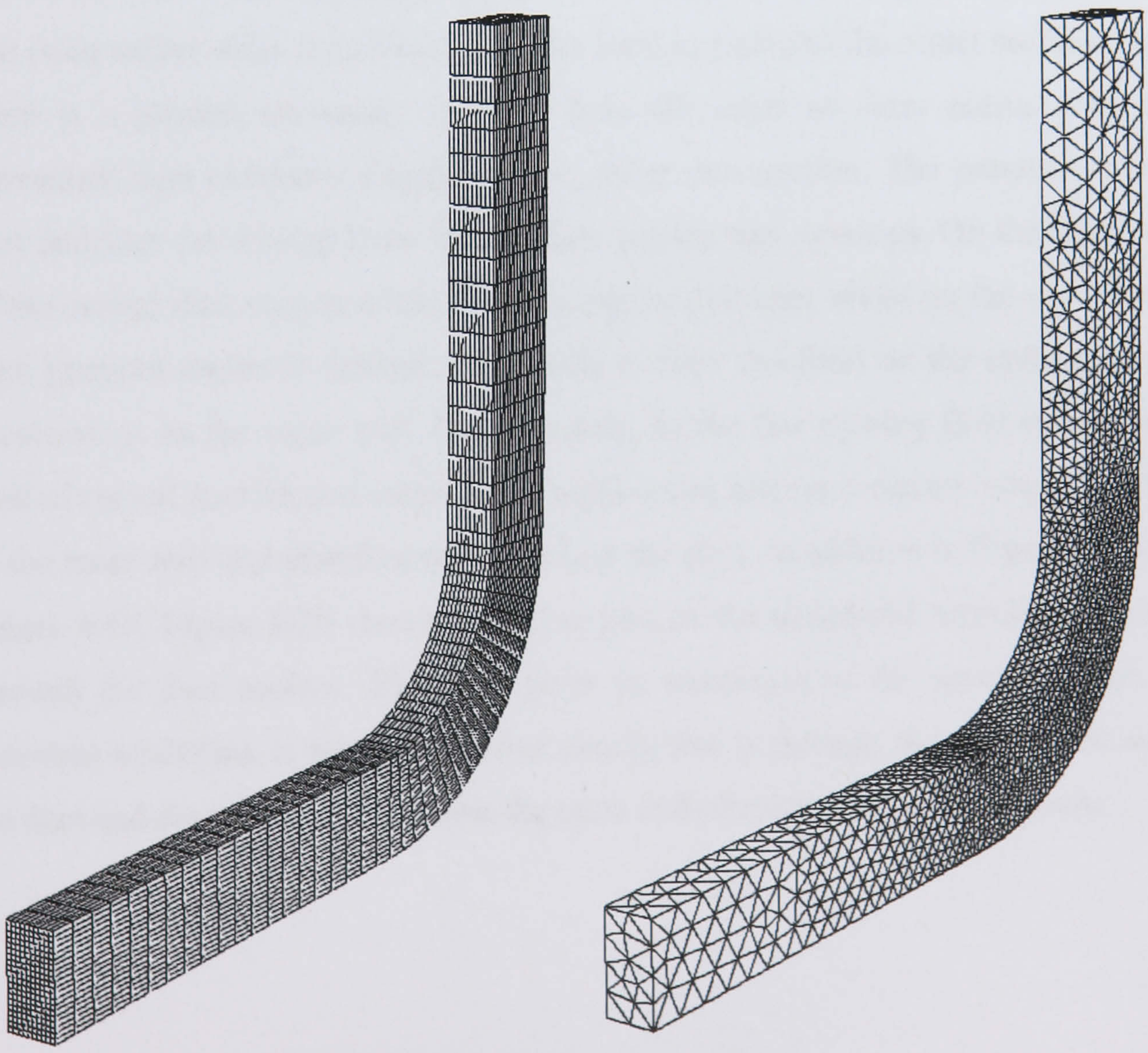


Figure 6.17: Structured and Unstructured Duct Meshes

As the turbulent computational geometry is identical to the geometry used for the equivalent laminar test case the same six meshes were used to calculate solutions. These were three unstructured tetrahedral meshes containing 49193 CV, 116415 CV and 148835 CV and three hexahedral grids containing 20x10x60 CV, 28x14x84 CV and 40x20x120 CV. An example of both forms of grid is shown in Figure 6.17. Calculations are performed on both types of grid to facilitate a comparison of the accuracy of solutions calculated using structured and unstructured meshes.

Figure 6.18 and Figure 6.19 show contour plots of the speed and pressure distributions along the centreline of the computational domain. These contour plots were created from data calculated using the 148835 CV unstructured tetrahedral mesh and second order convective modelling. In the inlet section of the duct the fully developed velocity profile specified at the inlet boundary is maintained. As the curved section of the duct begins the fluid accelerates and decelerates along the inner and outer radius walls respectively. As this section ends and the outlet section begins there is a general movement of fluid from the inner to outer radius wall, this movement then continues throughout the outlet duct section. The pressure contour plot indicates the driving force for the flow pattern that develops. On the inner wall of the curved duct section a low pressure region develops whilst on the outer wall a high pressure region is formed. This tends to slow the fluid on the outer wall and accelerate it on the inner wall. Consequently, as the fast moving fluid on the inner wall of curved duct section enters the straight outlet section it cannot remain attached to the inner wall and therefore moves across the duct. In addition to Figure 6.18 and Figure 6.19, Figure 6.20 shows a contour plot of the calculated turbulent viscosity through the duct section. This plot gives an indication of the area in which the turbulent modelling is most active, and clearly this is through the curved section of the duct and downstream of it where the most complicated flow patterns occur.

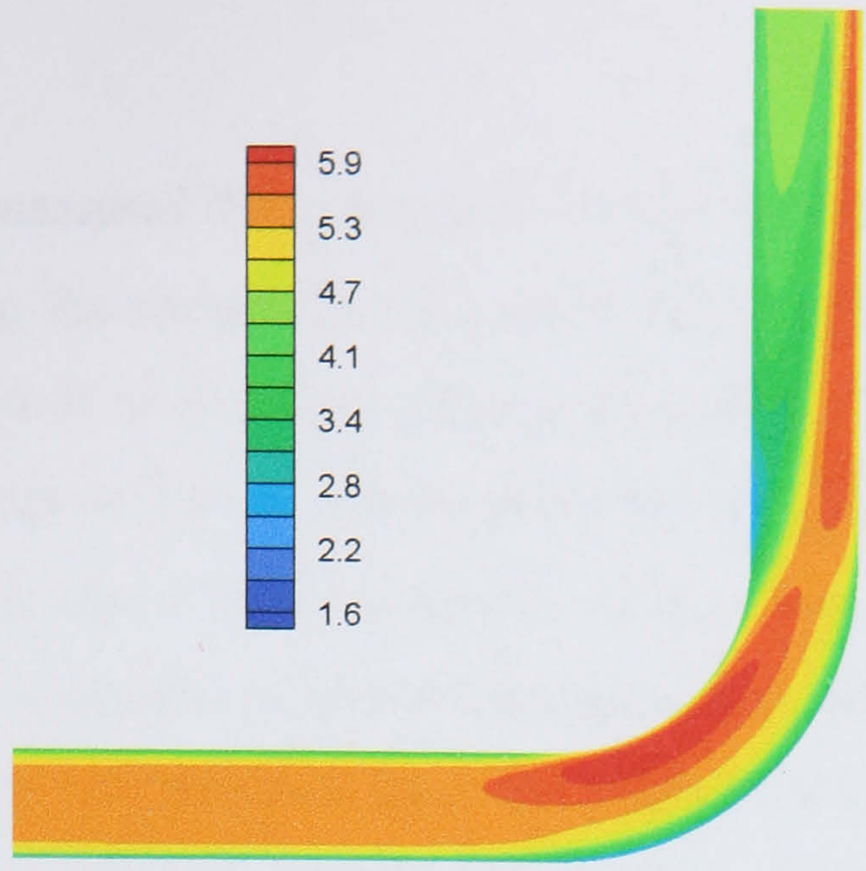


Figure 6.18: Speed (m/s) Contours in the Duct

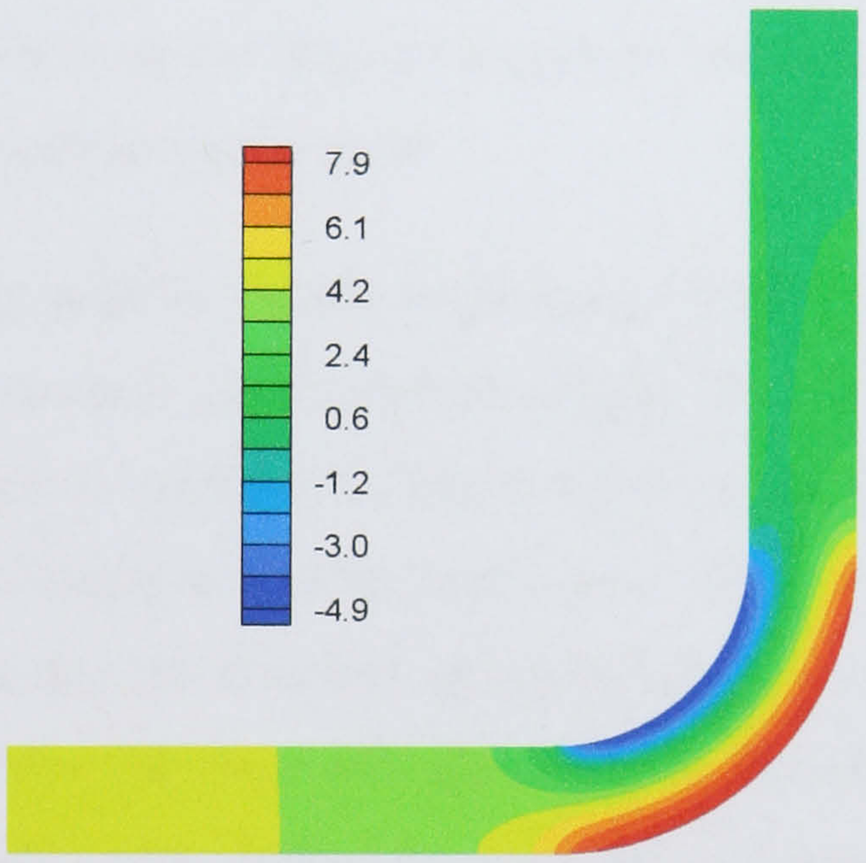


Figure 6.19: Relative Pressure (Pa) Contours in the Duct

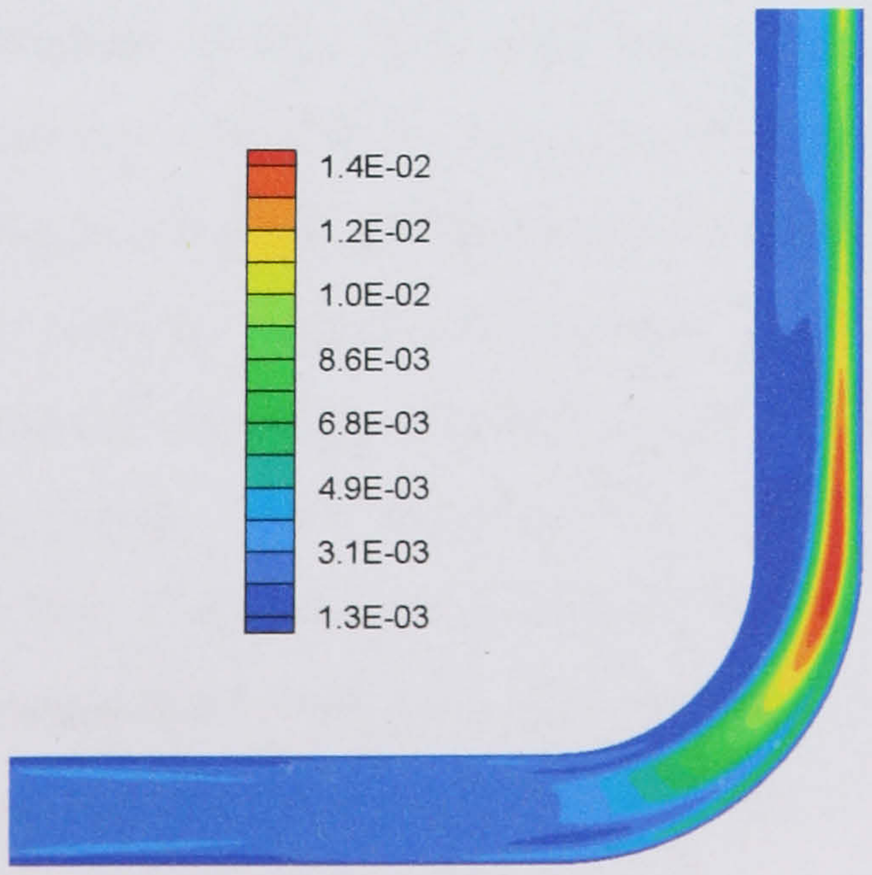


Figure 6.20: Contours of Turbulent Viscosity (kg/ms) in the Duct

Kliafas and Holt[103] measured the streamwise velocity and turbulent kinetic energy at four locations within the curved duct section, θ=0°, θ=15°, θ=30° and θ=45°, all shown in Figure 6.16. It is useful, in addition to velocity data, to have data on the turbulent kinetic energy as it gives a direct method of assessing the accuracy of the *k-ε* model rather than via a velocity profile as has been done previously. The experimental data is compared to numerical results in Figure 6.21 and Figure 6.22. The profiles presented are calculated using the 148835 CV unstructured grid and the 40x20x120 CV structured mesh. Results for first, second and third order convective models are presented for the unstructured grid whereas the results from the structured mesh are constrained to just second order convective modelling. Results using the other convective models on the structured grid are not presented as they show the same patterns as the unstructured results.

Consider the velocity profiles for the unstructured results. Clearly, the second and third order models produce more accurate results than the first order convective modelling. This differs to previous turbulent results presented in this chapter where in general all the convective models tested have been of similar accuracy. The obvious difference is that the duct test case is a three-dimensional geometry whilst previous test cases have only been two-dimensional. Therefore, it seems likely that the influence of high order modelling is more significant in predicting three-dimensional flows than it is in two-dimensional problems. In general, the results calculated using the high order models predict the experimental data well in the core of the duct. Nevertheless in the near wall region as in previous test cases discrepancies exist that are inevitably caused by the inadequacies of the wall function approach. Comparison between the unstructured and structured data shows that the discretisation strategy performs equally well on either grid. This is in contrast to the laminar test case where differences were more apparent. However closer examination of the laminar results reveals that the discrepancies existed in downstream locations where experimental data was not measured by Kliafas and Holt[103]. Without this experimental data a meaningful comparison between the two grid types could not be made in these locations and therefore is not attempted here.

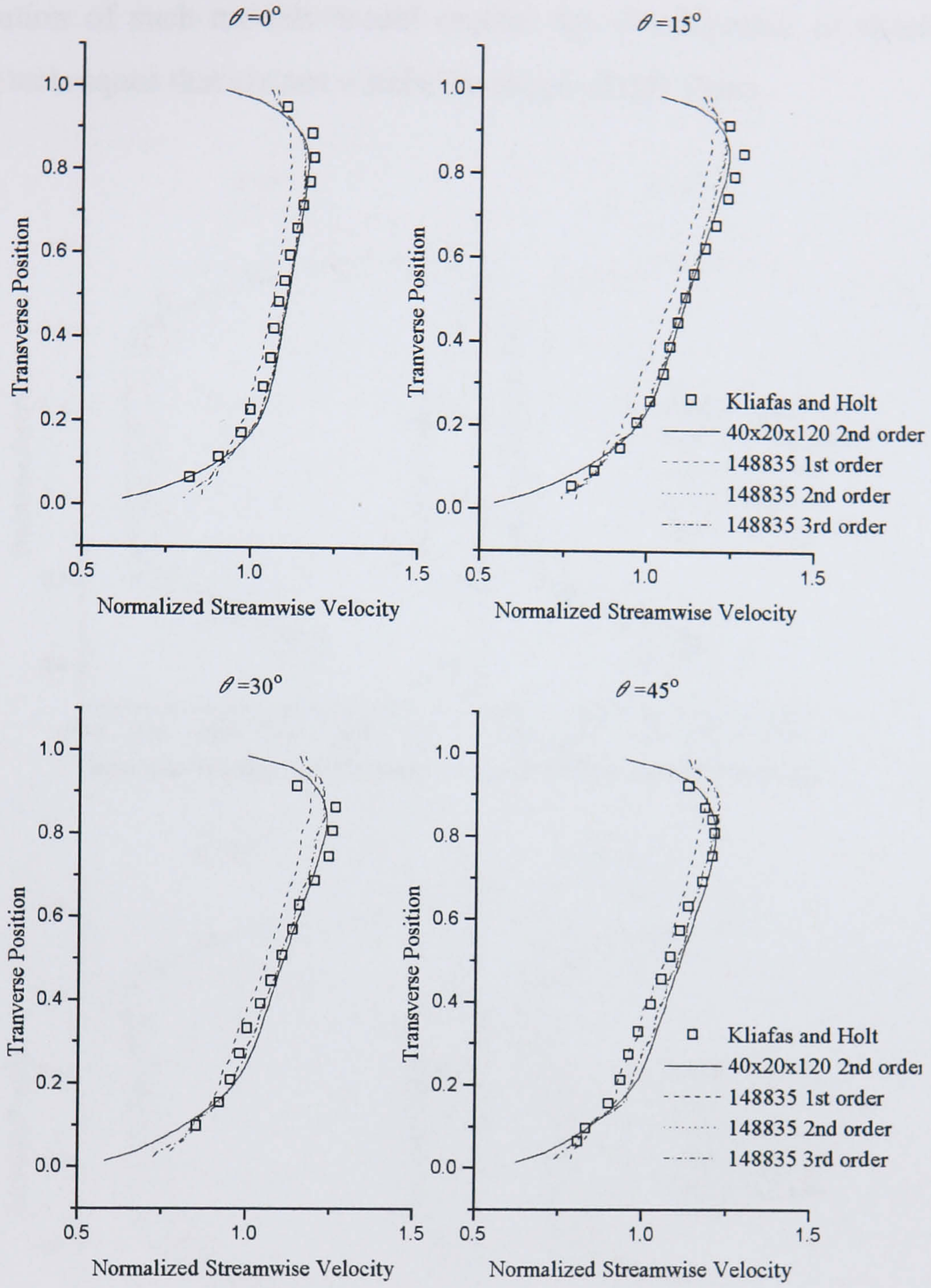


Figure 6.21: Normalised Streamwise Velocity Profiles at Four Longitudinal Locations

The profiles of turbulent kinetic energy presented in Figure 6.22 form a similar pattern in terms of their accuracy to the velocity profiles shown in Figure 6.21. Again, the first order convective model fails to be as accurate as the higher order schemes and the results from the structured and unstructured grids are of comparable accuracy. The kinetic energy at the core of the duct is by in large in good agreement with Kliafas and Holt's[103] experimental data whilst in the near wall region, as with the velocity profiles, there are discrepancies with the experimental data. Again this is an inevitable product of using wall functions in this region rather than one of the more accurate near wall turbulence models. However as in two dimensions the

implementation of such models would require the development of specialist grid refinement techniques that are not within the scope of this thesis.

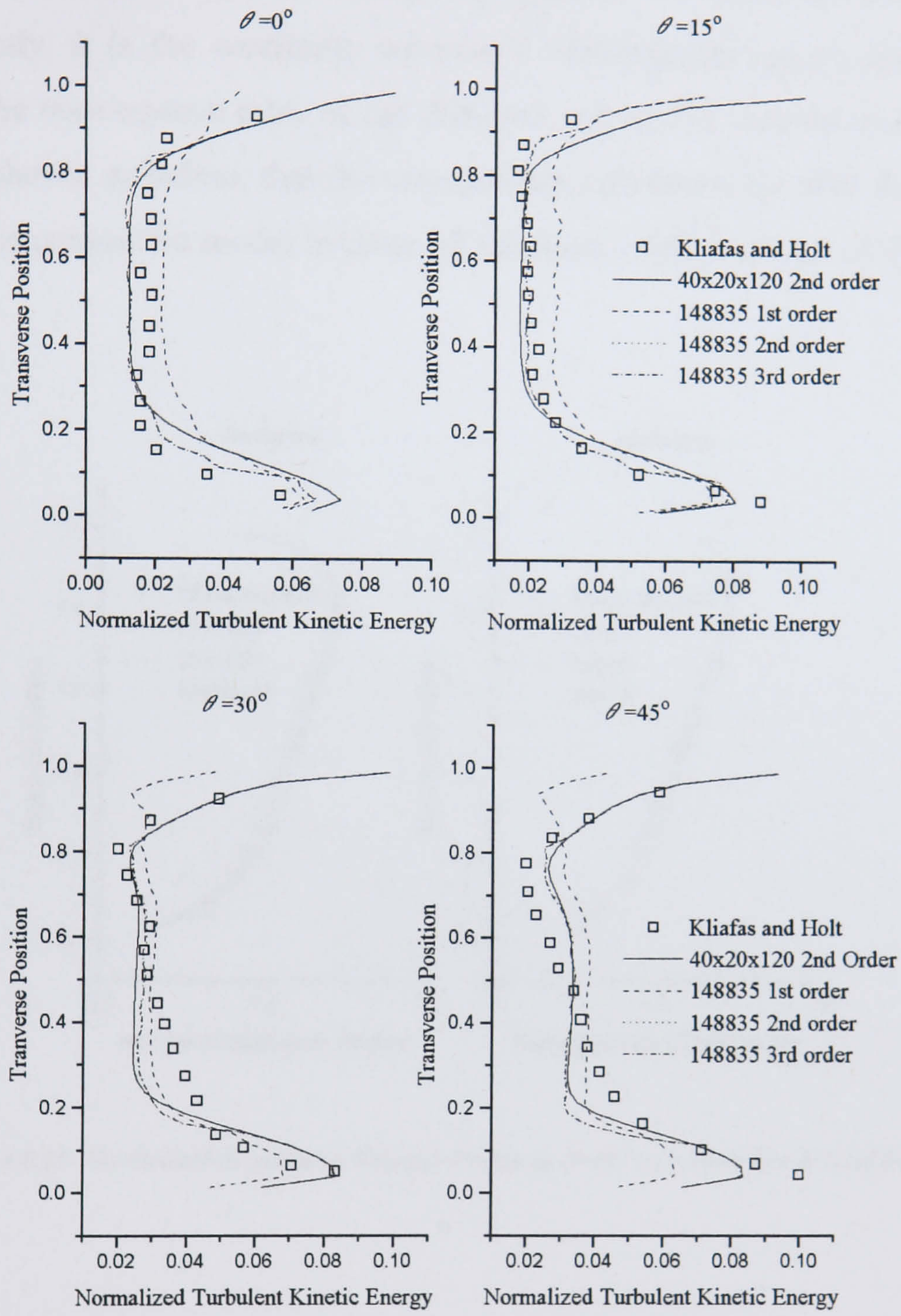


Figure 6.22: Normalised Turbulent Kinetic Energy Profiles at Four Longitudinal Locations

Figure 6.23 shows the streamwise velocity profile at θ=45° calculated using second order convective modelling for the three unstructured and structured grids respectively. Clearly in both cases a grid independent solution has been achieved and therefore the comparisons between different sized structured and unstructured meshes made earlier are quite valid. Finally, Figure 6.24a shows the convergence history for all six governing equations whilst Figure 6.24b shows a comparison of the convergence of the different convective models. The convergence histories shown in

Figure 6.24a were both calculated using the 148835 CV grid and second order convective modelling. Clearly the continuity equation has the slowest convergence rate closely followed by the transport equation for turbulent kinetic energy. Consequently, it is the continuity equation's convergence history that is used to compare the convergence rates of the different convective models in Figure 6.24b. This plot shows, as before, that the convergence rate decreases with the increase in order of the convective model because of the more explicit nature of the governing equations.

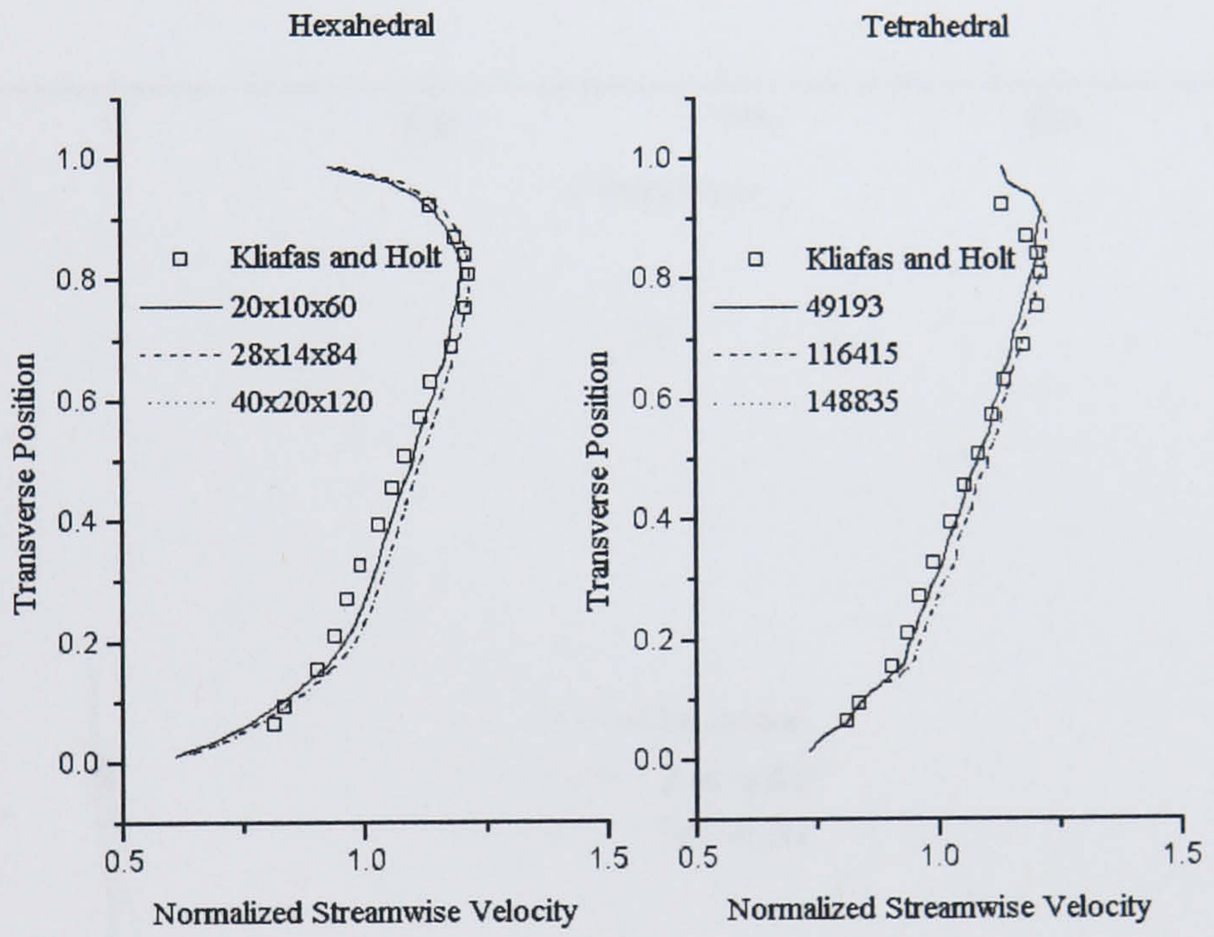
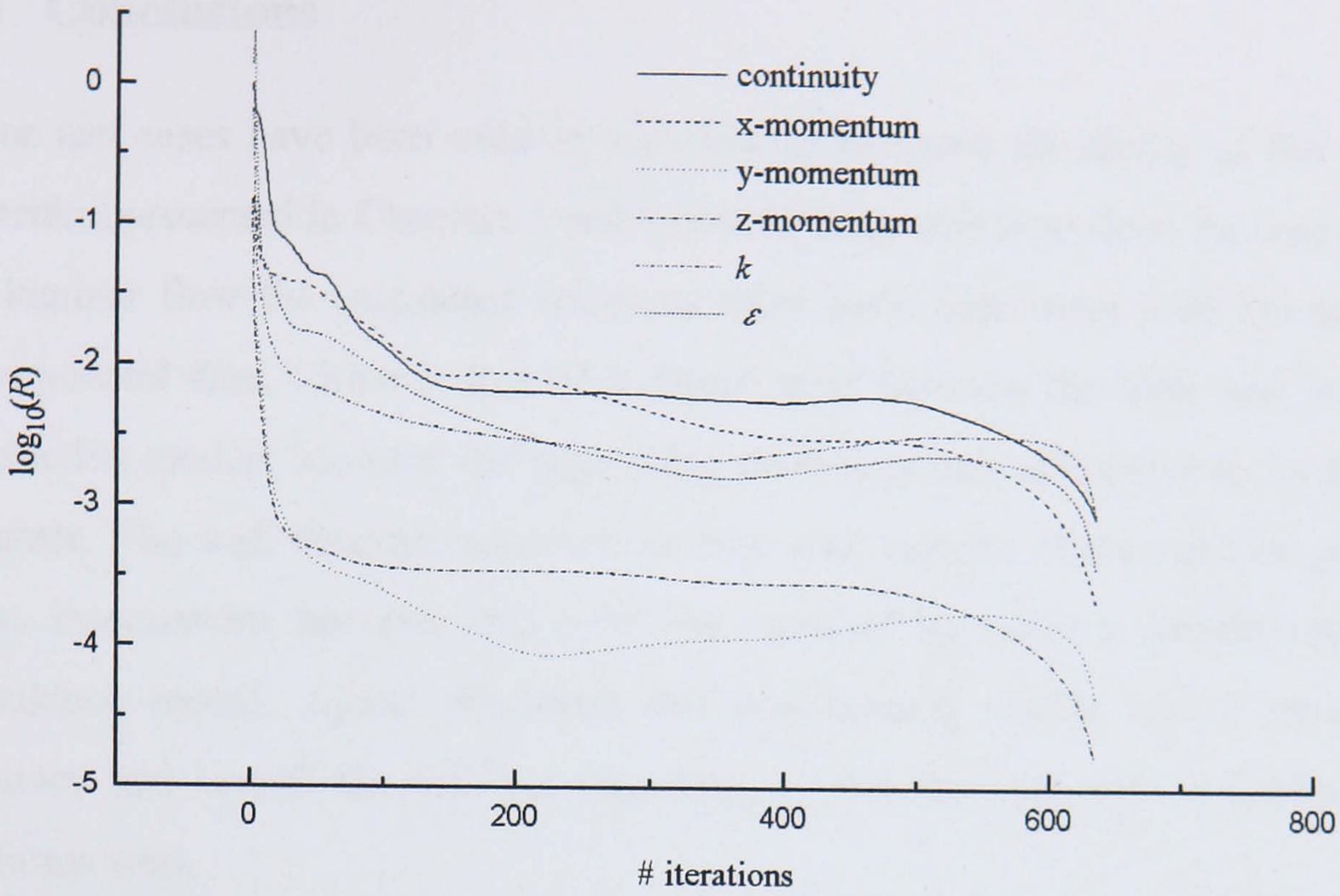
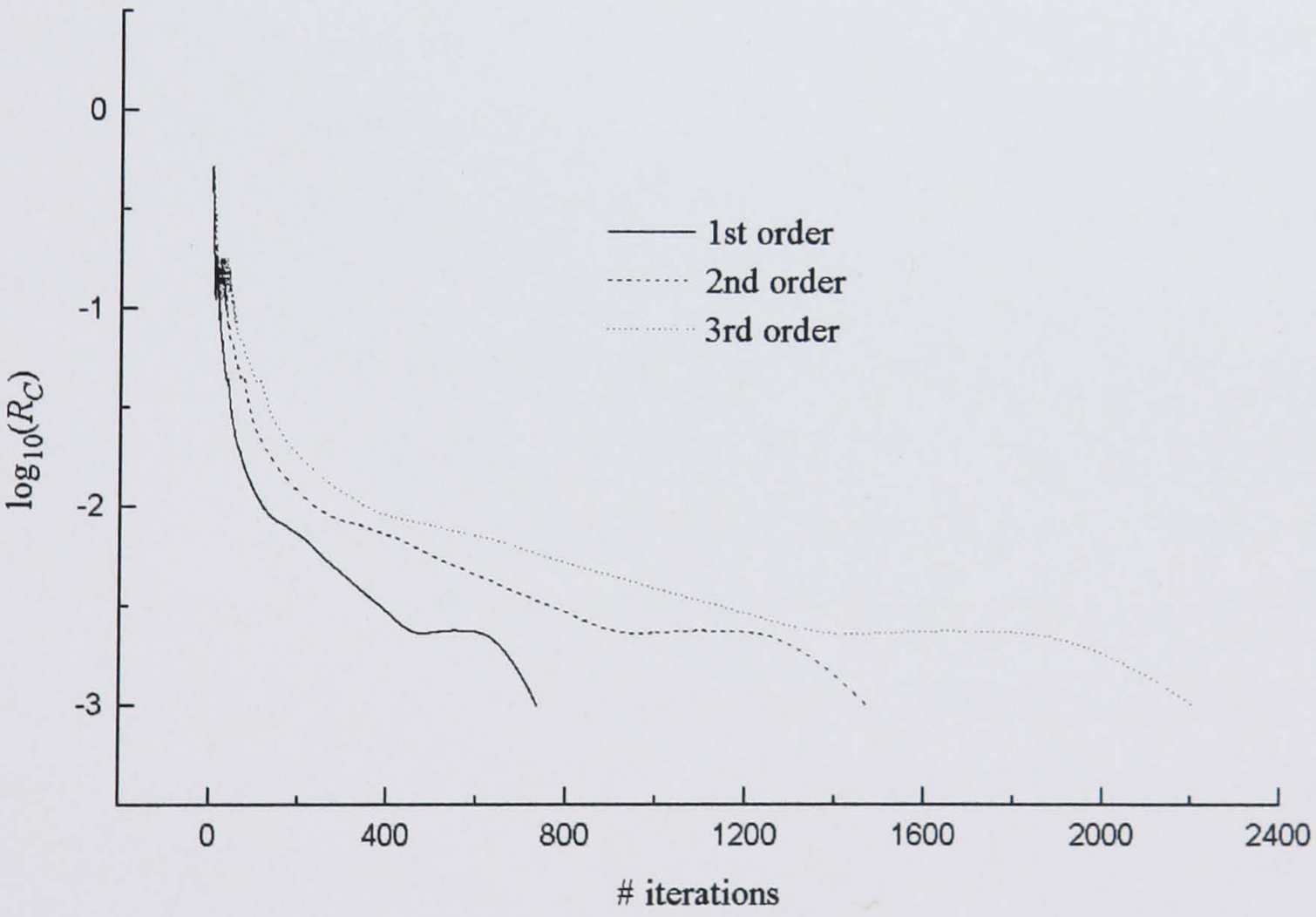


Figure 6.23: Normalised Streamwise Velocity Profile at θ=45° as a Function of Grid Fineness



(a)



(b)

Figure 6.24: Duct Convergence Histories (a) Comparison between Governing Equations (b) Comparison between Convective Models

6.5 Conclusions

Three test cases have been used in this chapter to assess the ability of the solution algorithm presented in Chapters 3 and 4 at modelling turbulent flow. As was the case for laminar flow the calculated solutions show good agreement with the measured experimental data. There is less of a discrepancy between the high and low order convective models however the high order models overall still prove to be the most accurate. The wall function approach in near wall regions of the solution produces some inaccuracies however this could be rectified by using a suitable near wall turbulence model. Again structured and unstructured results are of comparable accuracy and overall the solution algorithm, as was the case with in laminar flow, performs well.

CHAPTER 7

7 Conclusions

The aim of this thesis was to present a numerical method that solved the incompressible Navier-Stokes equations on unstructured meshes. A close examination of the open literature revealed that whilst numerical methods using finite element techniques to solve this problem are widely used, the body of work utilising finite volume techniques was less developed. Nevertheless a small amount of research had been carried out using finite volume methods, but scope existed to add to the ideas already presented and move this field of research forward. An area identified here as suitable for further work were those finite volume methods that were steady, cell centred and based upon the SIMPLE[15] algorithm. On structured meshes this form of finite volume method had been shown to be particularly effective, when high order upwind convective models formed part of the discretisation. However, on unstructured meshes only a second order discretisation of the convective term had been achieved. Clearly this presented an obvious area of research that should be explored and to that end, here, a convective model has been developed that can have first, second, third or a higher order of accuracy. In addition previous steady, cell centred, SIMPLE based, finite volume methods had been restricted to calculating laminar flows. Here the standard k-ε model with wall functions has been introduced into the discretisation in order that turbulent flows can also be modelled effectively. Finally, it was found that standard methods of solving systems of algebraic equations did not allow the numerical algorithm to find solutions within a reasonable time frame, and therefore a form of algebraic multigrid was developed to alleviate this problem.

It is useful now to discuss the different areas of research presented in this thesis and examine how successful they have been at solving the particular problem they were designed to address. To begin with consider the area of convective modelling. The laminar results presented in Chapter 5 showed clearly that high Reynolds number can

have a drastic influence on the accuracy of the first order convective modelling because of the effect of numerical diffusion. As would be expected the second and third order models do not suffer from this difficulty however what is perhaps more unexpected is that they produce quite similar results. Table 5.1 shows that third order results are in fact more accurate however the difference is not large. As detailed in Chapter 5 this can be accounted for in two ways. Firstly the difference between a first and second order model will always be larger than the difference between a second and third order model because of the nature of the polynomials involved. However in addition to this the diffusion term in the momentum equation and the discretisation of the continuity equation is only second order accurate and therefore these will play a dominant role over the third order accuracy of the convective term. In conclusion then whilst a second order accurate convective model shows obvious benefits over it's first order counterpart the benefits of using third order accuracy are not so great. Nonetheless the fact that a method for third or higher order discretisation has been presented here demonstrates that higher order discretisations are possible on unstructured grids despite the lack of line structure within the mesh.

The results from the turbulent test cases presented in Chapter 6 show less significant differences between all the convective models. As highlighted in this chapter this can be attributed to a number of reasons including the effective lowering of the Reynolds number of a given flow because of the addition of artificial viscosity into the momentum equations. This in turn obviously lowers the amount of numerical diffusion present in a solution and therefore similar results are achieved to those of the low Reynolds number laminar test cases. This is particularly apparent in the two-dimensional test cases although in three dimensions there is still a discernable improvement in accuracy between the first and higher order convective models. However, the overall accuracy of the solution is probably more influenced by the dominate accuracy level of discretisation within the governing equations, highlighted above, and the effectiveness with which the k-ε models the characteristics of turbulent flow. Inaccuracy in the turbulent model is clearly shown in the near wall regions in all the turbulent test cases and for all the convective models. As is well known wall functions are not the ideal solution to near wall turbulence but it is beyond the scope of this thesis to either develop a new model to solve this problem

or test the multitude of existing turbulence models that may offer a solution. However, a framework has been put in place that will allow a large majority of time averaged based turbulence models to be used, with little alteration to either the solution algorithm or the discretisation strategy, and therefore it is envisaged that more accurate turbulence modelling could be achieved.

Comparisons of accuracy, in Chapters 5 and 6, were not only drawn between the different convective models, but also between solutions calculated on structured and unstructured grids. This is an important comparison to make as the relative inaccuracy of some unstructured algorithms in comparison to their structured counterparts, has contributed to unstructured methods not being widely adopted, despite their obvious advantages. The results shown here demonstrate that comparable accuracy between structured and unstructured algorithms can be achieved if an appropriate discretisation is adopted. The key to achieving this is dealing successfully with the non-orthogonal nature of an unstructured grid. This causes extra terms to form part of the discretisation of the governing equations. Clearly, the evaluation of gradients in the convective term is one example of this that has been handled successfully here through the use of a least square formulation. However other terms such as the non-orthogonal part of the diffusion term have not been analysed to ascertain whether the discretisation used here is sufficiently accurate. Consequently it is likely that further improvements in accuracy of unstructured calculations can be made as more analysis of different discretisations takes place.

Finally, consider the application of algebraic multigrid to unstructured meshes presented in Chapter 4. This idea was developed because standard methods of solving large systems of algebraic equations, such as Gauss-Seidel iteration, proved to be too slow when applied to the pressure correction equation. Clearly the results shown in Chapter 4 and subsequent chapters demonstrate that this scheme is a great success and forms an invaluable part of the overall solution algorithm. The scheme does have one drawback though and that is the amount of data that it is necessary to store for the algorithm to work efficiently. The data takes two forms, topological information about the succession of grids that the algorithm relies upon, and solution data calculated on those grids. The time constraints placed upon the research

presented here limited the amount of effort that could be expended on optimising data storage. Therefore further work would undoubtedly yield a more efficient data storage scheme and thereby lessen the impact the use of the multigrid scheme has on the overall memory requirements of the solution algorithm. However, this improvement would not alter the fundamental principles of the scheme that are presented here.

In conclusion then, three main areas have been identified where the work presented here, has added significantly to the body of knowledge relating to solving the incompressible Navier-Stokes equations on unstructured meshes. These were the use of high order convective modelling, the development of an algebraic multigrid scheme and the inclusion of a turbulence model within the solution algorithm. Clearly, only a limited number of test cases have been used to assess the accuracy of the solution algorithm and therefore the first area of future work should be to use the model on a wider variety of test cases. These should include fluid dynamics problems from all areas of engineering as one of the principle reasons for developing algorithms on unstructured meshes is their ability to be applied to a wide variety of problems and therefore this should be tested. Also, testing the model on a broad range of problems should allow the conclusions of this thesis to be confirmed and possibly highlight other advantages and disadvantages that are inherent to the algorithm presented here. After an extensive period of testing it should also be possible to develop the algorithm further. Improving the efficiency of the multigrid data storage is one possible development area as is investigating fully the accuracy of the discretisation of other terms in the governing equations, as was done for the convective term here. Also, the area of turbulence modelling could be significantly improved. Whilst the k-ε turbulence model has undoubtedly become the workhorse of turbulent flow calculations it is by no means perfect. New turbulence models are being developed all the time and inevitably these will provide greater levels of accuracy and therefore should be included within solution algorithms.

Considering now a slightly broader, picture the algorithm developed here is part of a family of methods that at some point should be compared. There is the obvious comparison to be made between finite volume and finite element methods but even within the limits of finite volume based schemes there is scope for comparison.

Hopefully this would lead to a greater understanding of the merits of the different schemes and possibly help to form a generic unstructured method that will become a standard for the solution of the incompressible Navier-Stokes equations. Outside of the actual solution algorithms there is also plenty of scope for further work on mesh generation, whether it be coupling the solution algorithm and the mesh generation process through adaptive refinement, or working on hybrid meshes that combine the best elements of structured and unstructured grids. Finally solution algorithms of all types are beginning to form part of integrated engineering design methods utilising inverse design and optimisation ideas. This type of CFD is really in it's infancy and there are many areas within this field which work as not even begun. However accurate, geometrically flexible solution algorithms underpin this research and hence the relevance of the work carried out here.

APPENDIX A

Vector Identities

$$(\mathbf{a}\times\mathbf{b})\times\mathbf{c}=(\mathbf{a}.\mathbf{c})\mathbf{b}-(\mathbf{b}.\mathbf{c})\mathbf{a} \qquad (A.1)$$

$$\nabla\times\nabla\phi=0 \qquad (A.2)$$

$$\nabla.(\nabla\times\mathbf{F})=0 \qquad (A.3)$$

$$\nabla.(\phi\mathbf{F})=\phi\nabla.\mathbf{F}+\mathbf{F}.\nabla\phi \qquad (A.4)$$

$$\nabla\times(\phi\mathbf{F})=\phi\nabla\times\mathbf{F}+(\nabla\phi)\times\mathbf{F} \qquad (A.5)$$

$$\nabla\times(\mathbf{F}\times\mathbf{G})=(\mathbf{G}.\nabla)\mathbf{F}-(\mathbf{F}.\nabla)\mathbf{G}+\mathbf{F}(\nabla.\mathbf{G})-\mathbf{G}(\nabla.\mathbf{F}) \qquad (A.6)$$

$$\nabla.(\mathbf{F}\times\mathbf{G})=\mathbf{G}.(\nabla\times\mathbf{F})-\mathbf{F}.(\nabla\times\mathbf{G}) \qquad (A.7)$$

$$\nabla(\mathbf{F}.\mathbf{G})=\mathbf{F}\times(\nabla\times\mathbf{G})+\mathbf{G}\times(\nabla\times\mathbf{F})+(\mathbf{F}.\nabla)\mathbf{G}+(\mathbf{G}.\nabla)\mathbf{F} \qquad (A.8)$$

$$(\mathbf{F}.\nabla)\mathbf{F}=(\nabla\times\mathbf{F})\times\mathbf{F}+\nabla\left(\frac{1}{2}\mathbf{F}^2\right) \qquad (A.9)$$

$$\nabla^2\mathbf{F}=\nabla(\nabla.\mathbf{F})-\nabla\times(\nabla\times\mathbf{F}) \qquad (A.10)$$

APPENDIX B

The Divergence Theorem

Let the region V be bounded by a simple closed surface S with unit outward normal **n**. Then

$$\int_S \mathbf{F}.\mathbf{n}dS = \int_V \nabla.\mathbf{F}dV \qquad (B.1)$$

In suffix notation, and using the summation convention, this takes the form

$$\int_S F_j n_j dS = \int_V \frac{\partial F_j}{\partial x_j} dV \qquad (B.2)$$

There are many identities which may be derived from the divergence theorem. The identity

$$\int_S \phi\mathbf{n}dS = \int_V \nabla\phi dV \qquad (B.3)$$

is particularly valuable, and may be written

$$\int_S \phi n_j dS = \int_V \frac{\partial \phi}{\partial x_j} dV \qquad (B.4)$$

The following are immediate consequences

$$\int_S F_i n_j dS = \int_V \frac{\partial F_i}{\partial x_j} dV \qquad (B.5)$$

$$\int_S T_{ij} n_j dS = \int_V \frac{\partial T_{ij}}{\partial x_j} dV \qquad (B.6)$$

$$\int_S u_i v_j n_j dS = \int_V \frac{\partial}{\partial x_j}(u_i v_j) dV \qquad (B.7)$$

Other identities derivable from the divergence theorem include

$$\int_S \mathbf{F} \times \mathbf{n} dS = -\int_V \nabla \times \mathbf{F} dV \qquad (B.8)$$

$$\int_S \mathbf{n}.\nabla \phi dS = \int_V \nabla^2 \phi dV \qquad (B.9)$$

$$\int_S \phi \frac{\partial \psi}{\partial n} dS = \int_V (\phi \nabla^2 \psi + \nabla \phi . \nabla \psi) dV \qquad (B.10)$$

$$\int_S \left(\phi \frac{\partial \psi}{\partial n} - \psi \frac{\partial \phi}{\partial n} \right) dS = \int_V (\phi \nabla^2 \psi - \psi \nabla^2 \phi) dV \qquad (B.11)$$

REFERENCES

[1] F. Thomasset. *Implementation of Finite Element Methods for Navier- Stokes Equations*. Springer-Verlag, New York, 1981.

[2] M.D. Gunzburger and R. Nicolaides (eds.). *Incompressible Computational Fluid Dynamics: Trends and Advances*. Cambridge University Press, New York, 1993.

[3] R. Lohner. *Applied CFD Techniques: An Introduction Based on Finite Element Methods*. Wiley & Sons, Chichester, 2001.

[4] L. Davidson. A Pressure Correction Method for Unstructured Meshes with Arbitrary Control Volumes, *International Journal for Numerical Methods in Fluids*. 22: 265-281, 1996.

[5] S.R. Mathur and J.Y. Murthy. A Pressure-Based Method for Unstructured Meshes. *Numerical Heat Transfer Part B*. 31: 195-215, 1997

[6] M. Thomadakis and M. Leschziner. A Pressure Correction Method for the Solution of Incompressible Viscous Flows on Unstructured Grids. *International Journal for Numerical Methods in Fluids*. 22: 581-601, 1996.

[7] G.K. Despotis and S. Tsangaris. Fractional Step Method for the Solution of the Incompressible Navier-Stokes Equations on Unstructured Meshes. *International Journal for Numerical Methods in Fluids*. 20: 1273-1288, 1995.

[8] Y. Jiang and A.J. Przekwas. Implicit Pressure Based Incompressible Navier-Stokes Equations Solver for Unstructured Meshes. *In 32nd Aerospace Sciences Meeting*. Reno, NV, USA, AIAA 94-0305, 1994

[9] B. de Foy and W. Dawes. Unstructured Pressure-Correction Solver based on a Consistent Discretisation of the Poisson Equation. *International Journal for Numerical Methods in Fluids*. 34: 463-478, 2000.

[10] J.K. Watterson. A Pressure Based Flow Solver for the Three Dimensional Navier-Stokes Equations on Unstructured and Adaptive Meshes. *In 25th AIAA Fluids Dynamics Conference*. Colorado Springs, CO, USA, AIAA 94-2358.

[11] J.M. Weiss and W.A. Smith. Preconditioning Applied to Variable and Constant Density Flows. *AIAA Journal*. 33(11): 2050-2057, 1995.

[12] W.K. Anderson, R.D. Rausch and D.L. Bonhaus. Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids. *Journal of Computational Physics*. 128: 391-408, 1996.

[13] C.W. Hirt and J.L. Cook. Calculating Three Dimensional Flows around Structures and Over Rough Terrain. *Journal of Computational Physics*. 10: 324-340, 1972.

[14] F.H. Harlow and J.E. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*. 8: 2182-2189, 1965.

[15] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, U.S.A., 1980

[16] A.J. Chorin. A Numerical Method for Solving Incompressible Viscous Flow Problems. *Journal of Computational Physics*. 2: 12-26, 1967.

[17] B.P. Leonard. A Stable and Accurate Convective Modelling Procedure based on Quadratic Upstream Interpolation. *Computer Methods in Applied Mechanics and Engineering*. 19: 59-98, 1979.

[18] A. Jameson, W. Schmidt and E. Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes. *AIAA-81-1259*, 1981.

[19] O.C. Zienkiewicz and R. Taylor. *The Finite Element Method*. McGraw and Hill, New York, 1988.

[20] C.M. Rhie and W.L. Chow. Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation. *AIAA Journal*. 21(11): 1525-1532, 1983.

[21] J.P. Van Doormal and G.D. Raithby. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer*. 7: 147-163, 1984.

[22] R.I. Issa. Solution of the Implicitly Discretised Fluid Flow Equations by Operator Splitting. *Journal of Computational Physics*. 62: 40-65, 1986.

[23] D.S. Jang, R. Jetli and S. Acharya. Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure Velocity Coupling in Steady Flow Problems. *Numerical Heat Transfer*. 19: 209-228.

[24] B. P. Leonard and J. E. Drummond. Why you should not use Hybrid and Power-Law or related Exponential Schemes for Convective Modelling - There are much better Alternatives. *International Journal of Numerical Methods in Fluids*. 20: 421-442, 1995.

[25] M.A. Leschziner. Practical Evaluation of Three Finite Difference schemes for the Computation of Steady State Recirculating Flow. *Computer Methods in Applied Mechanics and Engineering.* 23: 293-312, 1980.

[26] T. Han, J.A.C. Humphrey and B.E. Launder. A Comparison of Hybrid and Quadratic Upstream Differencing in High Reynolds Number Elliptic Flow. *Computer Methods in Applied Mechanics and Engineering.* 29: 81-95, 1981.

[27] A. Pollard and A.L. Siu. The Calculation of some Laminar Flows using Various Discretisation Schemes. *Computer Methods in Applied Mechanics and Engineering.* 35: 293-313, 1982.

[28] T. Hayase, J.A.C. Humphrey, and R. Greif. A Consistently Formulated QUICK Scheme for Fast and Stable Convergence using Finite-Volume Iterative Calculation Procedures. *Journal of Computational Physics*. 98: 108-118, 1992.

[29] B.P. Leonard. A Survey of Finite Differences of Opinion on Numerical Muddling of Incomprehensible Defective Confusion Equation. *Paper in ASME, Applied Mechanics Division*. Winter Annual Meeting, 1979.

[30] M. Atias, M. Wolfshtein and M. Israel. Efficiency of Navier-Stokes Solvers. *AIAA Journal*. 15: 263-266, 1977.

[31] P.H. Gaskell and A.K.C. Lau. Curvature Compensated Convective Transport: SMART, a New Boundedness Preserving Algorithm. *International Journal of Numerical Methods in Fluids*. 8: 617-641, 1988.

[32] B.P. Leonard, Simple High Accuracy Resolution Program for Convective Modelling of Discontinuities. *International Journal of Numerical Methods in Fluids*. 8: 1291-1318, 1988.

[33] A. Harten. High Resolution Schemes for Hyperbolic Conservation Laws. *Journal of Computational Physics*. 49: 357-393, 1983.

[34] S.R. Chakravarthy and S. Osher. High Resolution Applications of the OSHER Upwind Scheme for the Euler Equations. *AIAA Paper 83-1943*. 1983.

[35] B. Van Leer. Towards the Ultimate Conservative Difference Scheme. V. A Second Order Sequel to Godunov's Method. . *Journal of Computational Physics*. 32: 101-136, 1979.

[36] B. Van Leer. Towards the Ultimate Conservative Difference Scheme. II. Monotinicity and Conservation Combined in a Second Order Scheme. . *Journal of Computational Physics*. 14: 361-370, 1974.

[37] M. S. Darwish. A New High Resolution Scheme based on the Normalised Variable Formulation. *Numerical Heat Transfer Part B*. 24: 353-371, 1993.

[38] J. Zhu and W. Rodi, A Low Dispersion and Bounded Convection Scheme. *Computer Methods in Applied Mechanics and Engineering*. 92: 87-96, 1991.

[39] F.M. White. *Viscous Fluid Flow*. McGraw and Hill, New York, 1991.

[40] O. Reynolds. On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Phil. Trans. Roy. Soc. London Ser. A*. 186: 123-164, 1895.

[41] J. Boussinesq. Essai sur la theorie des courantes (Essay on the theory of water flow). *Memoires Academie de Science (Paris)*. 23(1): 1-680, 1877.

[42] L. Prandtl. Uber die ausgebildete Turbulenz. *Z. Angew. Math. Mech.* 5: 136-139, 1925.

[43] D.B. Spalding. A Single Formula for the Law of the Wall. *J. Appl. Mech.* 28: 455-457, 1961.

[44] F.H. Clauser. The Turbulent Boundary Layer. *Adv. In Appl. Mech.* 4: 1-51, 1956.

[45] B.S. Baldwin and H. Lomax. Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow. *AIAA Paper 78-257.*

[46] P.S. Granville. Baldwin Lomax Factors for Turbulent Boundary Layers in Pressure Gradients. *AIAA J.* 25: 1624-1627, 1987.

[47] D.E. Coles. The Law of the Wake in the Turbulent Boundary Layer. *J. Fluid Mech.* 1: 191-226, 1956.

[48] C.J. Chen and S.Y. Jaw. *Fundamentals of Turbulence Modeling*. Taylor and Francis, Washington, U.S.A., 1998.

[49] P.A. Libby. *Introduction to Turbulence*. Taylor and Francis, Washington, U.S.A., 1996.

[50] B.E. Launder and D.B. Spalding. *Lectures in Mathematical Models of Turbulence*. Academic Press, London, England, 1972.

[51] K.Hanjalic and B.E. Launder. A Reynolds stress model of turbulence and its application to thin shear. *Journal of Fluid Mechanics*. 52(4): 609-638, 1972.

[52] B.E. Launder and D.B.Spalding. The Numerical Computation of Turbulent Flows. *Computer Methods in Applied Mechanics and Engineering*. 3: 269-289, 1974.

[53] W.P. Jones and B.E. Launder. The Calculation of Low Reynolds Number Phenomena with a Two Equation Model of Turbulence. *Interna. J. Heat. Mass Transfer.* 15: 301-314, 1972.

[54] B.E. Launder and B.I. Sharma. Application of the Energy Dissapation Model of Turbulence to the Calculation of Flow Near a Spinning Disc. *Letters in Heat and Mass Transfer.* 1: 131-138, 1974.

[55] V.C Patel, W. Rodi and G. Scheurer. Turbulence Models for Near Wall and Low Reynolds Number Flows: A Review. *AIAA J.* 23: 1308-1319, 1984

[56] W. Rodi. *Turbulence Models and their Applications in Hydraulics.* Brookfield Publishing, Brookfield, Vt. 1984.

[57] K. Hanjalic and B.E. Launder. A Reynolds Stress Model of Turbulence and its Application to Thin Shear Flows. *J. Fluid. Mech.* 52: 609-638, 1972

[58] M.B. Abbott and B.R. Basco. *Computational Fluid Dynamics – An Introduction for Engineers*. Longman Scientific and Technical, Harlow, UK, 1989.

[59] A. Leonard. Energy Cascade in Large Eddy Simulation of Turbulent Fluid Flows. *Advances in Geophysics*, 18A: 237-248, 1974.

[60] D.C. Leslie and G.L. Quarini. The Application of Turbulence Theory to the Formulation of the Subgrid Modelling Procedures. *J. Fluid. Mech.* 97(Part 1): 65-91, 1979.

[61] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Rev. NWB* 93: 99, 1963.

[62] R. Lohner and P. Parikh. Three Dimensional Grid Generation by the Advancing Front Method. *Int. J. Num. Meth. Fluids.* 8: 1135-1149, 1988.

[63] P.L. George and H. Borouchaki. *Delaunay Triangulation and Meshing* Hermes, Paris, 1998.

[64] R. Lohner. *Applied CFD Techniques*. Wiley, Chichester, 2001.

[65] A. Bowyer. Computing Dirichlet Tessellations. *Comput. J.* 24(2): 162-167, 1981.

[66] D.F. Watson. Computing the n-dimensional Delaunay Tesselation with Application to Voronoi Polytopes. Comput. J. 24(2): 167-172, 1981.

[67] W.H. Frey. Selective Refinement a New Strategy for Automatic Node Placement in Graded Triangular Meshes. Int. J. Num. Meth. Eng. 24: 2183-2200, 1987.

[68] D.G. Holmes and D.D. Snyder. The Generation of Unstructured Triangular Meshes Using Delaunay Triangulation. *Numerical Grid Generation in Computational Fluid Dynamics*. Pineridge Press, Swansea, Wales, pp643-652, 1988.

[69] M. Delanaye and J.A. Essers. Finite volume scheme with quadratic reconstruction on unstructured adaptive meshes applied to turbomachinery flows. *Journal of Turbomachinery*. 119: 263-269, 1997.

[70] W.H. Press, B.P. Flannery, S.A. Teulosky and W.T. Vetterling *Numerical Recipes: The Art of Scientific Computing*.

[71] T.J. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. *27th AIAA Aerospace Sciences Meeting, Reno NV*. AIAA Paper 89-0366, 1989.

[72] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady state solutions, AIAA Paper 93-0880, 1993.

[73] S. Majumdar. The role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. *Numerical Heat Transfer*. 13: 125-132, 1998.

[74] M. Peric, *University of London Thesis*, 1985

[75] A.D. Gosman and F.J.K.Ideriah. Turbulence Paper. *Journal*, 1980.

[76] E. Blosch, W. Shyy and R. Smith. The role of mass conservation in pressure-based algorithms. *Numerical Heat Transfer Part B*. 24: 415-429, 1993.

[77] S.R. Mathur and J.Y. Murthy. Pressure boundary conditions for incompressible flow using unstructured meshes. *Numerical Heat Transfer Part B*. 32: 283-298, 1997.

[78] G.D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press, London, 1978.

[79] H.L Stone. Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations. *SIAM J. Num. Anal.* 5: 530-558, 1968.

[80] J. Ruge and K. Stuben. Efficient Solutions of Finite Difference and Finite Element Equations by Algebraic Multigrid. *Arbeitspap. GMD No. 89.* 1984.

[81] A. Brandt. Multi-Level Adaptive Solutions to Boundary-Value Problems. *Mathematics of Computation.* 31(138): 333-390, 1978.

[82] B.R. Hutchinson and G.D. Raithby. A Multigrid Method Based on the Additive Correction Strategy. *Numerical Heat Transfer.* 9: 511-537, 1986.

[83] R.D. Lonsdale. An Algebraic Multigrid Solver for the Navier-Stokes Equations on Unstructured meshes. *International Journal of Numerical Methods for Heat and Fluid Flow.* 3: 3-14, 1993.

[84] U. Ghia, K.N. Ghia and C.T. Shin. High Reynolds number solutions for incompressible flow using the Navier stokes equations and a multigrid method, *Journal of Computational Physics*. 48: 387-411, 1982.

[85] I. Demirdzic, Z. Lilek and M. Peric. Fluid flow and heat transfer test problems for non-orthogonal grids: benchmark solutions, *International Journal for Numerical Methods in Fluids.* 15: 329-354, 1992.

[86] C.W. Oosterlee, P. Wesseling, A. Segal and E. Brakkee. Benchmark solutions for the incompressible Navier-Stokes equations in general coordinates on staggered grids, *International Journal for Numerical Methods in Fluids.* 17: 301-321, 1993.

[87] B.F. Armaly, F.Durst, J.C.F. Pereira and B. Schonung. Experimental and theoretical investigation of backward facing step flow, *Journal of Fluid Mechanics*. 127: 473-496, 1983.

[88] M.K. Denham and M.A. Patrick. Laminar flow over a downstream-facing step in a two dimensional flow channel. *Trans. Inst. Chem. Engrs*. 52:361, 1974

[89] D.W. Etheridge and P.H. Kemp. Measurements of turbulent flow downstream of a backward facing step. *Journal of Fluid Mechanics*. 86: 545, 1978

[90] P. Vauschkuhn and V. Vasanta Ram. Die turbulente Grenzschicht hinter einem Ablosegebiet. *Z. Flugwiss*. 23: 1, 1975.

[91] P. Vauschkuhn and V. Vasanta Ram. Die turbulente Grenzschicht unmittelbar hinter dem Wiederanlegen einies Ablosegebietes. *Z. angew. Math. Mech.* 55: T166, 1975.

[92] O.R. Burggraf. Analytical and numerical studies of the structure of steady separated flows, *Journal of Fluid Mechanics.* 244: 113, 1966.

[93] J.A. Humphrey, A.M.K. Taylor and J.H. Whitelaw. Laminar flow in square duct of strong curvature. *Journal of Fluid Mechanics.* 83: 509-527, 1977.

[94] M. Rosenfeld, D. Kwak and M. Vinokur. A fractional step solution method for he unsteady incompressible Navier-Stokes equation in generalized coordinate system. *Journal of Computational Physics.* 94: 102-137, 1991.

[95] J.P. Jessee and W.A. Fiveland. A cell vertex algorithm for the incompressible Navier-Stokes equations on non-orthogonal grids. *International Journal for Numerical Methods in Fluids*. 23; 271-293,1996.

[96] S.K. Choi, H.Y. Nam, Y.B. Lee and M.Cho. An efficient three dimensional calculation procedure for incompressible flows in complex geometries. *Numerical Heat Transfer Part B*. 23: 387-400, 1993

[97] S.E. Rogers, D. Kwak and M. Vinokur. Steady and unsteady solutions of the incompressible Navier-Stokes equations. *AIAA Journal*. 29:603-610, 1991.

[98] N. Gregory and C.L. O'Reilly. Low Speed Aerodynamic Characteristics of NACA 0012 Airfoil Section, including the Effects of Upper-Surface Roughness simulating Hoar Frost. *National Physical Laboratory, Teddington, England*. Aero Report 1308, 1970.

[99] J.C. Vogel and J.K. Eaton. Combined Heat Transfer and Fluid Dynamics Measurements Downstream of a Backward Facing Step. *Journal of Heat Transfer*. 107: 922-929, 1985.

[100] D.M. Driver and H.L. Seegmiller. Features of a Reattaching Turbulent Shear Layer in a Divergent Channel Flow. *AIAA Journal*. 23: 163-171, 1985.

[101] J.A.C. Humphrey, J.H. Whitelaw and G.Yee. Turbulent Flow in a Square Duct with Strong Curvature. *J. Fluid Mech.*, 103: 443-463, 1981.

[102] A.M.K.P. Taylor, J.H. Whitelaw and M. Yianneskis. Curved Ducts with Strong Secondary Motion: Velocity Measurements of Developing Laminar and Turbulent Flow. *ASME J. Fluids Eng.*, 104: 350-359, 1982.

[103] Y. Kliafas and M. Holt. LDV Measurements of a Turbulent Air-Solid Two-Phase Flow in a 90° Bend. *Exp. Fluids*, 5: 73-85, 1987.